

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE

DE LA MAITRISE EN GÉNIE ÉLECTRIQUE

PAR

ABDELWAHED KHAILA

OBSERVATEUR DE FLUX POUR LA COMMANDE
VECTORIELLE DE LA MACHINE ASYNCHRONE : CONCEPTION
ET IMPLANTATION

JUILLET 2003

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Remerciements

L'aboutissement de ces années passées au sein du laboratoire de recherche en génie électrique de l'Université du Québec à Trois-Rivières m'a permis de bien préparer ma maîtrise dans des meilleures conditions. Qu'il me soit ici donné l'opportunité d'avoir une pensée pour toutes les personnes qui, chacune à leur façon, m'ont accompagné et guidé tout au long de ce chemin.

Je tiens tout d'abord à remercier sincèrement tous les membres qui ont participé à ce travail:

- Mr **Cheriti Ahmed**, professeur à l'UQTR, qui a proposé ce sujet.
- Mr **Ghouily Jamel**, professeur de l'Université de Moncton de m'avoir m'encadré durant mon stage d'été en 2002.
- Je tiens à remercier aussi l'ensemble du personnel de l'UQTR et du département Génie Électrique.
- Les personnes de ma famille.
- Egalement, je tiens à remercier Mr **Barazouk Abdelfatah** et Mr **Sandali Abdelhalim**.

Résumé

Notre projet consiste à l'estimation de variables inaccessibles telles que le flux du rotor d'une machine asynchrone sans l'utilisation de codeurs incrémentaux, absous ou de tachymètres.

Nous avons choisi d'utiliser le filtre de Kalman étendu (EKF) pour l'estimation des paramètres de l'état des machines de forte et moyenne puissance, cette approche est basée sur la définition d'un critère qui exige moins de temps de calcul, est aussi robuste et précis.

Le filtre fonctionne donc en boucle ouverte et, l'ensemble machine-commande ne diverge pas.

En pratique, d'autres problèmes apparaissent tels que les problèmes numériques qui conduisent parfois à la divergence de l'estimateur.

Dans le cas du filtre de Kalman, il est possible de réduire les effets numériques par l'emploi d'algorithmes optimaux en se basant sur une structure matricielle telles que les racines carrées, les inverses, la triangularité et la diagonale.

L'objectif de ce mémoire est l'emploi d'un algorithme optimal qui permet la reconstruction du vecteur d'état (variables non mesurables) pour réaliser des commandes sans capteur mécanique. Cet algorithme va permettre de réduire le nombre d'opérations pour l'implémentation du filtre numérique et d'augmenter la stabilité de l'ensemble machine-commande afin d'accroître son utilisation dans les variateurs industriels de hautes performances.

Ce travail consiste donc à l'étude du modèle du système avec le filtre de Kalman optimal qui nécessite un DSP moins performant que son homologue classique à l'étape de l'implantation. En effet, si on désire garder la taille du problème raisonnable et encore obtenir plus de performances avec le minimum d'instructions (DSP raisonnable), nous avons choisi d'effectuer le travail avec un filtre de Kalman optimisé d'ordre 2 et un logiciel de simulation performant tel que le Visual C++.

Une analyse détaillée et qualitative de cet observateur de flux ainsi conçu a pu être menée à la fois en simulation et en implantation. Les performances obtenues sont intéressantes et ce pour un degré de complexité de l'observateur tout à fait acceptable.

L'évaluation des besoins pour une implémentation sur calculateur a été abordée en détails. La discréétisation de l'observateur, un système dynamique, a été réalisée en tenant compte du caractère des différents signaux électriques.

Liste des sigles et des symboles

Acronymes

CAO	Conception Assistée par Ordinateur
DSP	Digital Signal Processeur / μ -processeur dédié au traitement des signaux
FOC	Commande Vectorielle / Field Oriented Control
LFT	Transformation Linéaire Fractionnaire / Linear Fractional Transform
LFR	Représentation Linéaire Fractionnaire / Linear Fractional Representation
LMI	Inégalité Matricielle Affine / Linear Matrix Inequality
LPV	Linéaire à Paramètres Variants (Système LPV) / Linear Parameter Varying
LTI	Linéaire Stationnaire (Système LTI) /Linear Time Invariant
MAS	Machine asynchrone
MCC	Machine à courant continu
EKF	Filtre de Kalman étendu
KF	Filtre de Kalman
P	Variable de Laplace

Notations

Signaux et systèmes

• L_2	Ensemble des signaux de carrés sommables
• RL_2	Ensemble des matrices de transfert strictement propres, à coefficients réels et n'ayant pas de pas de pôles sur l'axe des imaginaires
• RL_∞	Ensemble des matrices de transfert propres, à coefficients réels et n'ayant pas de pas de pôles sur l'axe des imaginaires
• $\ T(s)\ _2$	Norme 2 du transfert ou de la matrice de transfert $T(s)$
• $\ T(s)\ _\infty$	Norme infinie du transfert ou de la matrice de transfert
• L	Transformée de Laplace monolatère
• Z	Transformée en z
• S	Variable de Laplace ou opérateur de dérivation suivant le contexte
• q	Opérateur avance
• $\begin{bmatrix} A & B \\ C & D \end{bmatrix} :$	Matrice système associée à la représentation d'état $\begin{cases} x = Ax + Bu \\ y = Cx + Du \end{cases}$

Matrices

• I_k, I	Matrice identité de dimension k ou définie par le contexte
• P'	Transposée de la matrice P
• $\text{rang}(P)$	Rang de la matrice P
• A, B, C	Matrice transition, de commande et de mesure en continue
• A_d, B_d, C	Matrice transition , de commande et de mesure en discret
• F	Matrice transition linéarisée
• H	Matrice de mesure linéarisée
• X_a	Vecteur d'état étendu
• K	Gain de Kalman
• Q	Matrice de variance-covariance de bruit d'état
• R	Matrice de variance-covariance de bruit de mesure
• W	Bruit d'état
• V	Bruit de mesure
• $P[k+1/k]$	Matrice de variance-covariance d'erreur de prédiction
• $P[k+1/k+1]$	Matrice de variance-covariance d'erreur d'estimation
• $X[k+1/k]$	Vecteur d'état prédit
• $X[k+1/k+1]$	Vecteur d'état estimé stator et rotor

- $[R_s], [R_r]$ Matrices diagonales des résistances statoriques et rotoriques
- $[L_s], [L_r]$ Matrices diagonales des inductances statoriques et rotoriques
- $[M_{sr}]$ Matrice des inductances mutuelles cycliques stator, rotor

Ensembles

- \mathbb{R} Ensemble des nombres réels
- $\mathbb{R}^{n \times m}$ Ensemble des matrices de dimension $n \times m$ à coefficients réels
- \mathbb{C} Ensemble des nombres complexes
- $\mathbb{C}^{n \times m}$ Ensemble des matrices de dimension $n \times m$ à coefficients complexes
- \mathbb{N} Ensemble des nombres entiers
- $S_\Sigma = \text{Co}\{S_{\Sigma 1}, \dots, S_{\Sigma 2^d}\}$ Ensemble convexe polytopique défini par ses sommets $S_{\Sigma i}$
- $S_\Sigma = \left\{ S_\Sigma = \sum_{i=1}^{2^d} \alpha_i S_{\Sigma i} : \sum_{i=1}^{2^d} \alpha_i = 1, \alpha_i \geq 0 \right\}$
- j, i $\sqrt{-1}$

Machine Asynchrone

• T_1	Couple de charge
• $i_{sd,qx}$	Courants statoriques dq
• $v_{sd,qx}$	Tensions statoriques dq
• $\Phi_{rd,qx}$	Flux rotorique
• $V_s = [V_{sa}, V_{sb}, V_{sc}]^t$	Tensions statoriques
• $V_r = [V_{ra}, V_{rb}, V_{rc}]^t$	Tensions rotoriques
• $i_s = [i_{sa}, i_{sb}, i_{sc}]^t$	Courants statoriques
• $i_r = [i_{ra}, i_{rb}, i_{rc}]^t$	Courants rotoriques
• $\Phi_s = [\Phi_{sa}, \Phi_{sb}, \Phi_{sc}]^t$	Flux magnétiques au stator
• $\Phi_r = [\Phi_{ra}, \Phi_{rb}, \Phi_{rc}]^t$	Flux magnétiques au rotor
• L_{as}, L_{ar}	Inductances propres stator et rotor
• M_s	Inductance mutuelle entre deux phases stator
• M_r	Inductance Mutuelle entre deux phases rotor
• M	Inductance Mutuelle entre une phase stator et une phase rotor
• R_s, R_r	Résistances du stator et du rotor
• $L_s = L_{ps} - M_s$	Inductance cyclique du stator
• $L_r = L_{pr} - M_r$	Inductance cyclique du rotor
• $M_{sr} = \frac{3}{2} M$	Inductance Mutuelle cyclique entre stator et rotor

σ	$1 - \frac{M_{sr}^2}{L_s L_r}$	Coefficient de dispersion
T_ε		Matrice de rotation
$T_{3 \rightarrow 2}$		Transformation de Concordia (composante homopolaire non prise en compte)
T_{3-3}		Transformation de Concordia
T_e		Période d'échantillonnage
J		Moment d'inertie (moteur asynchrone + charge)
ω_m		Vitesse de rotation mécanique ramenée en grandeur électrique du rotor
Ω		Vitesse mécanique de rotation du rotor
ω_s		Pulsation électrique
ω_{sl}		Pulsation électrique du courant rotorique
p		Nombre de paires de pôles
θ		Position du rotor
C_{em}		Couple électromagnétique
C_r		Couple de charge
f		Frottement total de la machine et de la charge

Table des matières

<i>Remerciements</i>	II
<i>Résumé</i>	III
<i>Liste des sigles et des symboles</i>	V
<i>Table des matières</i>	XI
<i>Liste des figures</i>	XV
<i>Liste des tableaux</i>	XVI
1. Introduction générale	1
1.1. Objectifs	2
1.2. La problématique	3
1.2.1. Problématiques liées à la commande	3
1.2.2. Problématiques liées à l'observation	5
1.3. Les observateurs de flux	5
1.3.1. Introduction	5
1.3.2. L'observateur en boucle ouverte	6
1.3.2.1. Observateurs en boucle ouverte (tension statorique)	7
1.3.3. L'observateur de Luenberger	9
1.3.3.1. Principes	9
1.3.3.2. L'observateur de Verghese	10
1.3.3.3. L'observateur de Kalman	13
1.3.3.4. Observateur par mode glissants	14
1.3.3.5. L'observateur à grands gains	16
1.4. L'observateur de Flux linéaire paramètre variant (LPV)	19
1.5. Synthèse d'un observateur LMI linéaire à paramètres variants	20

1.5.1. Problème standard associé à un observateur de flux LPV	21
1.5.2. Réglage de la bande passante de l'observateur	23
1.5.3. Synthèse et analyse de l'observateur de flux LPV	23
1.5.3.1. Méthodologie de discréétisation de l'observateur	24
1.5.3.1.1. Introduction	24
1.5.3.1.2. Méthode proposée	25
Conclusion	29
 2. L'analyse du filtre de Kalman	 31
<i>Résumé</i>	31
2.1. Introduction	31
2.2. Principe du filtrage de Kalman	32
2.2.1. Généralités	32
2.2.2. Le filtre de Kalman	33
2.2.2.1. Équations du filtre de Kalman	33
2.2.3. Estimateur étendu (non linéaire)	35
2.2.3.1. Observabilité du système augmenté	37
2.2.3.1.1. Principe	37
2.2.3.1.2. Définition de l'observabilité globale	39
2.2.3.1.3. Définition de l'observabilité locale	39
2.3. Problèmes d'implantation du filtre	39
2.3.1. Introduction	39
2.3.2. Influence des erreurs numériques sur le comportement du filtre	40
2.3.3. Propagation des erreurs d'arrondi dans le filtre de Kalman	41
2.4. Les moyens de réglage du filtre de Kalman	43
Conclusion	44
 3. Méthodologie pour la reconstruction du flux rotorique d'une machine asynchrone	 45
<i>Résumé</i>	
3.1. Introduction	45

3.2. Modélisation du système considéré	46
3.2.1. Généralités	46
3.2.2. Mise en équations électriques et magnétiques	46
3.2.3. Transformation de Park	50
3.2.3.1. Modèle dans le plan diphasé($\alpha - \beta$) équivalent	50
3.2.4. Équations de Park de la machine asynchrone dans les axes dq	52
3.2.5. Système (d'état) non linéaire	54
3.2.6. Choix du référentiel	57
3.2.6.1. Lieux des pôles pour les trois repères	57
3.2.6.2. Choix du repère	61
3.3. Méthodologie de discréttisation du modèle continu de la machine asynchrone	62
3.3.1. Structure des matrices A_d et B_d	63
3.3.2. Modèle 1 : Développement limité à l'ordre 2 de e^{ATs} : Calcul de A_d et B_d	67
3.3.3. Modèle 2: Calcul de A_d par un développement limite à l'ordre 2 de e^{ATs} et calcul de B_d à un ordre supérieur	68
3.4. Augmentation de l'état	70
3.4.1. Linéarisation du modèle	70
3.4.1.1. Calcul de A_d et B_d par un développement limite d'ordre 1 de e^{ATs}	70
3.4.1.2. Simulation du filtre de Kalman d'ordre 1 de e^{ATs}	71
3.4.1.2.1. Résultats de la simulation	73
3.4.1.2.2. Calcul de A_d et B_d par un développement limite d'ordre 2 de e^{ATs}	75
3.4.1.2.3. Calcul des gains de correction	75
Conclusion	76

4. Amélioration de l'algorithme du filtre de Kalman	77
<i>Résumé</i>	
4.1. Introduction	77
4.2. Choix des matrices Q et R	77
4.2.1. Pré-réglage automatique du filtre de Kalman	77
4.2.1.1. Structure et réglage de R	77
4.2.1.2. Structure et réglage de Q	79
4.2.1.3. Interprétation des matrices et leur réglage de Q	79
4.2.1.3.1. Pré-réglage des degrés de liberté	80
4.2.1.3.2. Simulation du filtre de Kalman d'ordre 2 de e^{ATs} comparé à LMI	81
4.2.1.3.3. Analyse des performances du filtre	82
4.2.1.3.4. La stabilité	86
4.3. Algorithme simple et optimal du filtre de Kalman	86
4.3.1. Étape de prédiction	86
4.3.1.1. Prédiction des états	86
4.3.1.2. Calcul de la variance de l'erreur de prédiction $P[k+1/k]$	87
4.3.2. Étape de correction	89
4.3.2.1. Calcul des gains de correction $k[k+1]$	89
4.3.2.2. Étape d'estimation des états	89
4.3.2.3. Calcul de la variance de l'erreur d'estimation $P[k+1/k+1]$.	90
4.3.2.4. Coût algorithmique du filtrage de Kalman optimal	90
Conclusion	92
Conclusion générale	93
Bibliographies	95
<i>Annexe</i>	100
• Programme en visual C++ sur le filtre de Kalman d'ordre 2	100
• Programmes de simulation en Matlab/Simulink du filtre de Kalman classique stochastique d'ordre 1	121

Liste des Figures

Figure1.1 : Le couple électromagnétique et le couple de charge	3
Figure 1.2 : Modèle de commande lié au problème de commande vectorielle de la machine asynchrone	4
Figure1. 3 : Modèle d'observation pour la machine asynchrone	5
Figure1. 4: L'estimateur de modèle tension	8
Figure1. 5 : Flux estimé et flux réel	8
Figure1.6 : L'erreur d'estimation du flux	9
Figure1.7 : Schéma bloc d'un filtre de Kalman	14
Figure1.8 : Problème standard associé à la synthèse d'un observateur de flux LPV pour la MAS	22
Figure1.9 : Description LFT de l'observateur LPV	26
Figure1.10: Description LFT de l'observateur LPV discret	28
Figure 2.1 :Apport d'une "symétrisation " des matrices de variance_covariance d'erreur d'estimation	41
Figure 2.2 : Divergence temporaire du filtre de Kalman	42
Figure 3.1: Modèle de la machine asynchrone dans le repère de Park	49
Figure 3.2: Lieu des pôles dans un repère lié au champ tournant	58
Figure 3.3: Lieu des pôles dans un repère lié au stator	60
Figure 3.4: Lieu des pôles dans un repère lié au rotor	60
Figure 3.5: Modèle Simulink du filtre de Kalman	72
Figure 3.5: Modèle Simulink (machine asynchrone et le filtre de Kalman)	73
Figure 3.7 :Flux réel comparé au flux estimé	73
Figure 3.8 :Vitesse réelle comparée à la vitesse estimée	74
Figure 4.1 : Flux rotorique réel	81
Figure 4.2 : Structure Simulink de l'observateur de flux LMI LPV	82
Figure 4.3 : Comparaison des performances des observateurs Kalman et LMI	84

Liste des tableaux

Tableau 3.1: Nombre d'opérations pour le calcul de A_d et B_d dans le cas du modèle1	68
Tableau 3.2: Nombre d'opérations pour le calcul de A_d et B_d dans le cas du modèle2	69
Tableau 3.3: Erreurs absolues maximales sur les coefficients de B_d	70
Tableau 3.3: Erreurs absolues maximales sur les coefficients de A_d	70
Tableau 4.1: Nombre d'opérations pour le calcul du filtre optimal et brut	91

Chapitre 1

1. Introduction générale

Cette recherche vise à améliorer la robustesse et les performances dynamiques de la machine à induction utilisée dans un grand nombre d'applications industrielles telles que les manufactures. Ces industries s'utilisent de plus en plus ce type de machine que ce soit pour accroître l'efficacité énergétique ou optimiser et améliorer les contrôles des procédés. Il existe de nombreuses machines d'entraînement qui se distinguent les unes des autres par le type de commande et la technologie utilisée. La machine à induction se distingue parmi les autres par sa robustesse, son coût relativement faible et son entretien peu fréquent. Parallèlement, le développement progressif des processus numériques de signaux a rendu possible l'implantation, à moindre coût des lois de commande (commande vectorielle linéaire ou non linéaire).

Ces dispositifs ont permis de retrouver la souplesse de commande et la qualité de conversion obtenue avec la machine à courant continu.

La commande des machines électriques se base sur des mesures de grandeurs physiques qui doivent fournir une information suffisante en quantité et en qualité sur l'état de ces machines. La précision de ces mesures est primordiale pour obtenir le niveau de performances que réclame certaines applications industrielles.

Ces grandeurs d'état ou de sortie sont souvent difficilement accessibles pour des raisons techniques (flux) ou des problèmes de coût (vitesse, position,...).

En général, les grandeurs mesurées sont les courants statoriques et la position du rotor, obtenue à l'aide d'un capteur, ce qui impose son coût financier et augmente la complexité des montages. Donc, ce capteur engendre différents problèmes par sa nature physique.

Pour cela, notre capteur doit être précis et protégé contre les poussières et les chocs pour

éviter des oscillations de vitesse dues à une quantification trop importante de la position et pour assurer sa sûreté de fonctionnement dans des environnements industriels sévères. Même si on installe ce capteur entre le moteur et la charge, il exige un ventilateur qui est installé dans l'une des extrémités de l'arbre du moteur. Ce placement augmente l'éloignement de ces deux éléments, induisant donc un plus grand accouplement élastique. De plus, cette solution technique est relativement coûteuse puisque les accouplements et le capteur doivent supporter les coups de couple imposés par le moteur. Pour cela, il est nécessaire de chercher à supprimer les capteurs chaque fois que les performances de l'application le permettent en déduisant l'information de flux ou de la vitesse à partir des valeurs déjà mesurées (courant, tension).

1.1. Objectifs :

L'objet de ce travail est de proposer une solution d'ordre méthodologique pour aborder et résoudre les problèmes de commande d'un dispositif électrique, qui est la machine asynchrone dans notre cas. Il présente donc des exemples d'utilisation de la théorie des systèmes incertains et plus précisément les observateurs stochastiques pour l'estimation et l'analyse des problèmes concrets de génie électrique (supprimer les capteurs), c'est-à-dire la confrontation de ces problèmes au modèle de filtre de Kalman. Comme élément central de la présentation de ce travail, nous avons choisi de retenir l'observateur de Kalman optimal qui fait l'objet du chapitre 3 sur une machine discrétisé au second ordre. Nous pouvons effectuer l'exploitation des différentes symétries et anti-symétries présents dans la machine [1] pour l'observateur de Kalman, au contraire de l'observateur $H\infty$ LPV[2], qui ne contient pas de propriétés de symétrie évidente. En effet, le coût de calcul de l'observateur $H\infty$ LPV n'est pas donc optimisable (une opération trigonométrique est sensé équivalente à l'évaluation d'un polynôme d'ordre 5 [26]), les résultats présentés dans les paragraphes qui suivent illustrent et valident la méthode proposée.

Après un bref exemple introductif rappelant des observateurs en boucle ouverte au chapitre 1, nous abordons au chapitre 2, les outils pour l'analyse du filtre de Kalman, les principes généraux qui conduisent à l'obtention du modèle du filtre de Kalman et les problèmes d'implantation du filtre pour l'estimation du flux d'une machine.

Cette partie sert de base aux chapitres suivants, consacrés à l'application de l'observateur classique du filtre de Kalman traditionnel et l'observateur de Kalman optimal .

Au chapitre 3, nous abordons une méthodologie pour la reconstruction du flux rotorique, en se basant sur le modèle biphasé d'ordre 5 de la machine asynchrone triphasée.

Ceci se fait par la manipulation de la structure matricielle du système après la modélisation et la discrétisation de la machine. Nous sommes intéressés à choisir l'utilisation dans ce contexte du filtre de Kalman étendu (EKF). Des conclusions sont finalement tirées à l'issue de chaque chapitre permettant également plusieurs perspectives de prolongement du travail réalisé.

Le problème ne commence pas ici, mais plusieurs études ont été faites et me permettaient de savoir la situation actuelle: les solutions obtenues et les méthodes utilisées et envisagées jusqu'à maintenant. En effet, tout ceci me permettait d'améliorer le filtre de Kalman pour minimiser le coût de calcul, le temps d'exécution et la gestion de l'espace mémoire de la DSP utilisée.

1.2. La problématique

1.2.1. Problématiques liées à la commande

Depuis quelques décennies, la commande la plus appréciée est la commande vectorielle. Le problème de la commande de la machine asynchrone se résume à la régulation de la vitesse (Ω) ainsi que du module du flux (Fr) en dépit du couple de charge (T_1) (**Figure1.1**) et en ne mesurant que la vitesse et les courants ($i_{sd,q}$).

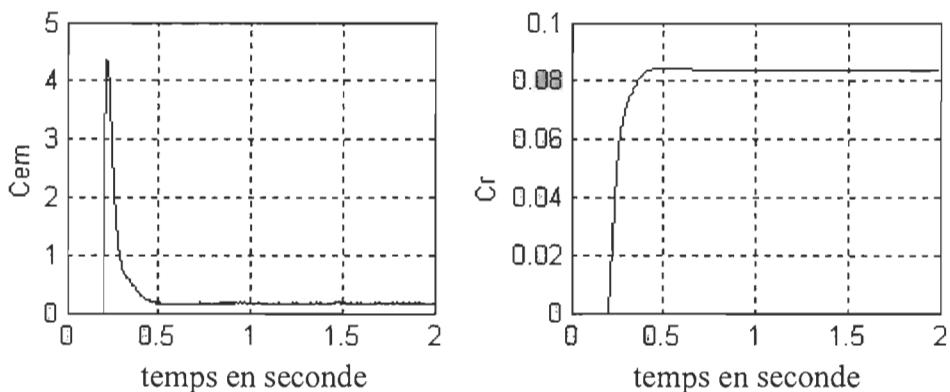


Figure1.1 : Couple électromagnétique et couple de charge

Le modèle de commande lié au contrôle vectoriel de la MAS est décrit par le schéma de la figure suivante (**Figure 1.2**). Quel que soit le référentiel dans le quel il est choisi de modéliser la machine, ce modèle de commande est non linéaire en plus d'être multivariable. Ces constatations prouvent l'impossibilité d'une approche PID.

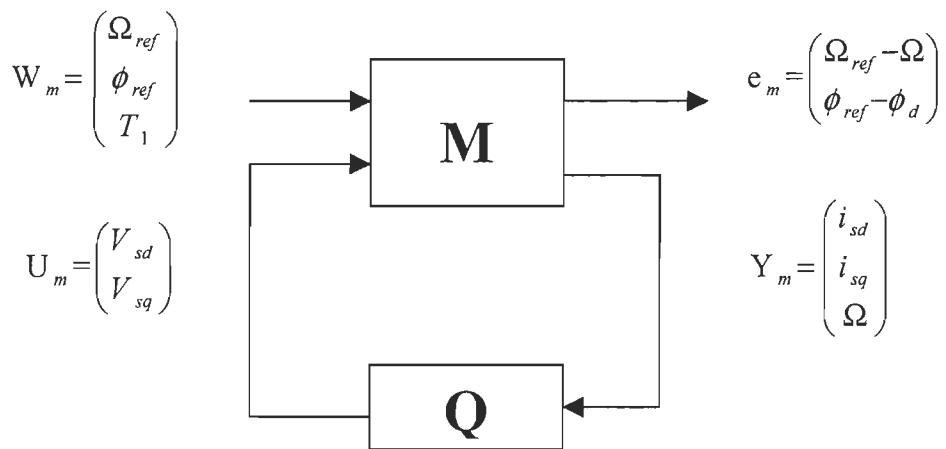


Figure 1.2 : Modèle de commande lié au problème de commande vectorielle de la machine asynchrone

Finalement, notons que la complexité de ce modèle de commande croît dans le cadre de la commande sans capteur en prenant en compte des contraintes thermiques ou mécaniques quelque soit l'approche envisagée.

Aujourd'hui, la tendance est le développement méthodologique et théorique des outils telle qu'il d'ingénieur CAO susceptibles de répondre aux besoins des applications industrielles telles qu'il est avec le filtre de Kalman optimal. Ces développements ont pour objectif la vulgarisation de certains résultats théoriques susceptibles de répondre aux besoins d'applications nécessitant l'optimisation des performances des régulations employées.

Toutefois, quel que soit le type d'approche envisagée, une bonne estimation du flux rotorique est primordiale.

1.2.2. Problématiques liées à l'observation

La complexité dans l'obtention d'un observateur est directement liée à la complexité du modèle considéré (exemple modèle non-linéaire d'ordre 6). En fait, un observateur idéal aurait les caractéristiques suivantes: simplicité, performance et robustesse vis-à-vis de certaines variations paramétriques. Actuellement, l'observateur le plus utilisé et aussi le plus simple à concevoir est un estimateur [4].

Ce dernier estime le flux rotorique uniquement à partir des mesures de courants et de la vitesse mécanique du rotor. La vitesse de convergence d'un tel estimateur n'est pas réglable et est imposée par la constante de temps rotorique. Comme dans le cas du problème de commande, le modèle d'observation (M) (figure1.3) est non-linéaire quel que soit le référentiel dans lequel la machine est modélisée. L'observateur fonctionne à temps continu, il est indispensable de procéder à une discrétisation de l'observateur en vue d'une future d'implémentation.

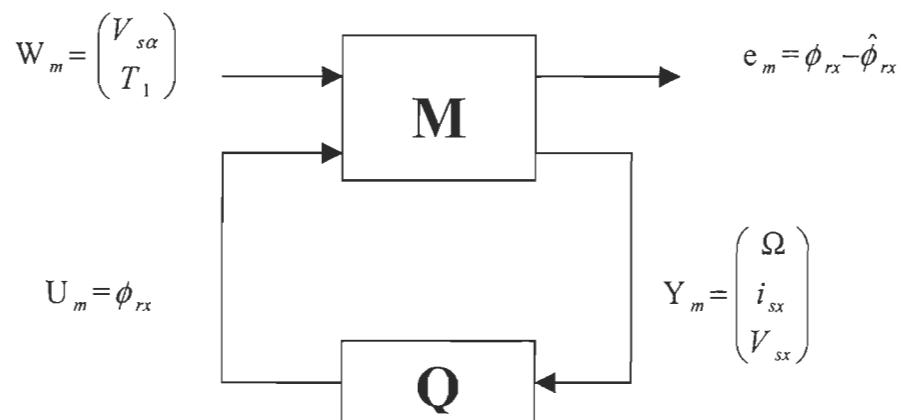


Figure1. 3 : Modèle d'observation pour la machine asynchrone

1.3. Les observateurs de flux

1.3.1. Introduction

Nous trouvons trois types d'approches dans la synthèse d'observateurs pour la machine asynchrone :

1. L'observateur en boucle ouverte qui est le plus ancien de tous les observateurs.

2. L'observateur de Verghese ou de Kalman qui reprend le principe de l'observateur de Luenberger [3] tout en restant intrinsèquement un système LPV.
3. L'observateur de type grands gains [4] ou modes glissants [5], troisième type d'approche, issu de la théorie des systèmes non-linéaires.

Ce sont ces trois points que nous nous proposons de développer au cours de cette section. Nous nous concentrerons plus particulièrement sur l'observateur proposé par Kalman en présentant certains résultats de simulations. Cet intérêt pour cet observateur est induit par le fait qu'il est dans sa catégorie le plus utilisé et le plus performant .

1.3.2. L'observateur en boucle ouverte

La façon la plus simple pour reconstruire une grandeur physique qui n'est pas facilement mesurable, c'est d'utiliser l'observateur en boucle ouverte. En fait, cette solution est souvent utilisée car elle exige moins de calcul algorithmique, ce qui permet d'envisager l'obtention d'une loi de commande d'un degré de complexité supérieur à celui de la commande vectorielle [6].

Pour les modèles qui seront considérés au (3.22), les estimateurs correspondants sont données par les expressions :

- dans le référentiel (α, β) [54].

$$\begin{cases} (\hat{\phi}_{r\alpha}) = \frac{1}{L_r} R_r \hat{\phi}_{r\alpha} - \omega_m \hat{\phi}_{r\beta} + \frac{M_{sr}}{L_r} R_r i_{s\alpha} \\ (\hat{\phi}_{r\beta}) = -\frac{1}{L_r} R_r \hat{\phi}_{r\beta} + \omega_m \hat{\phi}_{r\alpha} + \frac{M_{sr}}{L_r} R_r i_{s\beta} \end{cases} \quad (1.1)$$

- dans le référentiel (d, q) lié au rotor [54]:

$$(\hat{\phi}_{rd}) = -\alpha \hat{\phi}_{r\alpha} + \alpha M_{sr} i_{sd} \quad (1.2)$$

Les deux derniers estimateurs présentent presque la même complexité car celui décrit par l'équation (1.1) exige d'avoir une estimation de la position du flux rotorique pour réaliser la transformation de Park sur les mesures de courant et de tension de commande.

Finalement, il en résulte que l'estimateur dans le référentiel (d,q) à la structure [54] exprimé comme suit :

$$\begin{cases} (\hat{\phi}_{rd}) = -\alpha \hat{\phi}_{r\alpha} + \alpha M_{sr} i_{sd} \\ \dot{\hat{\phi}} = \omega + \alpha \frac{M_{sr}}{\hat{\phi}_{rd}} i_{sq} \end{cases} \quad (1.3)$$

On note que les équations (1.1) et (1.3) illustrent des estimateurs qui sont de part et d'autre simples, et donc relativement appréciables lorsqu'ils sont placés dans une boucle de commande qui les rendent les plus utilisés dans le domaine industriel.

Toutefois, un problème relativement important se pose, c'est celui de leur sensibilité paramétrique. En effet, la condition nécessaire et suffisante pour ne pas avoir de biais sur la reconstruction se résume à l'absence d'erreurs sur l'identification des paramètres de la machine asynchrone et sur les mesures des courants statoriques.

1.3.2.1. Observateurs boucle ouverte (tension statorique)

Il existe deux topologies de base d'observateur de flux en boucle ouverte qui font référence soit au modèle courant ou au modèle tension qui est le plus utilisé.

Le modèle de tension statorique utilise les mesures de tensions statoriques mais pas la vitesse rotorique, il est commandé d'implémenter l'orientation du champ direct sans mesurer la vitesse du rotor pour réduire le coût .

- Le flux rotorique est obtenu en utilisant les expressions:

$$\begin{cases} \Psi_{qdr} = L_r i_{qdr} + L_m i_{qds} \\ \Psi_{qds} = L_s i_{qds} + L_m i_{qdr} \end{cases} \quad (1.4)$$

- pour obtenir l'expression suivante du flux:

$$\hat{\Psi}_{qdr} = \hat{L}_r / \hat{L}_m \hat{\Psi}_{qds} - \hat{\delta} / (1 - \hat{\delta}) \hat{L}_m i_{qds} \quad (1.5)$$

Le bloc diagramme est réalisé en blocs Simulink et avec l'estimateur de modèle tension **figure(1.4)**:

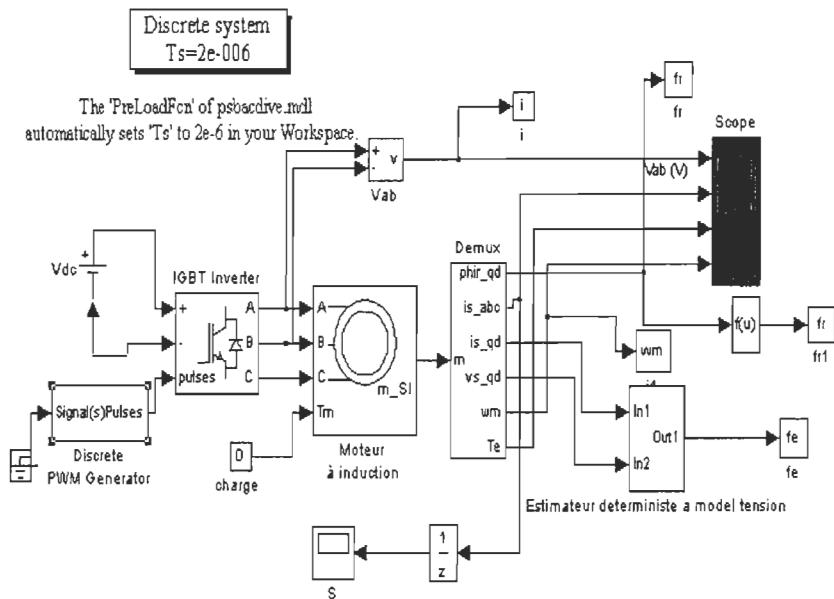


Figure1. 4: Blocs Simulink de l'estimateur de modèle tension

Les paramètres de l'estimateur utilisé ne sont jamais exactement corrects. L'évaluation des paramètres dépend de la performance de l'estimateur de flux.

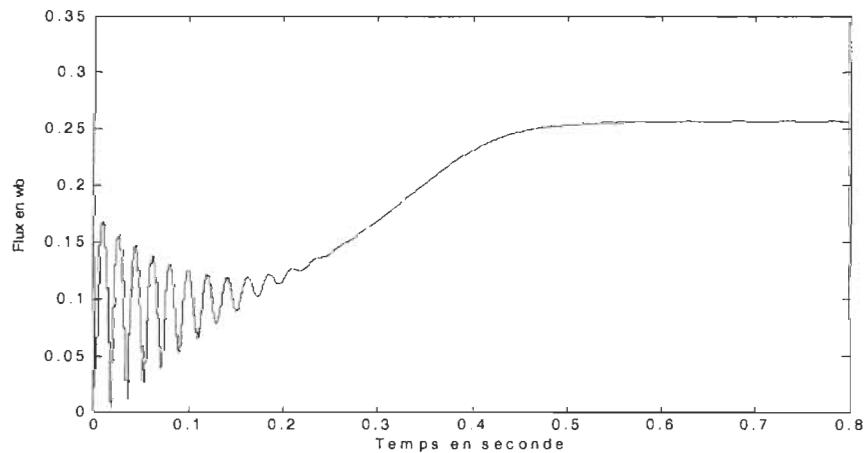


Figure1. 5 : Flux estimé et flux réel

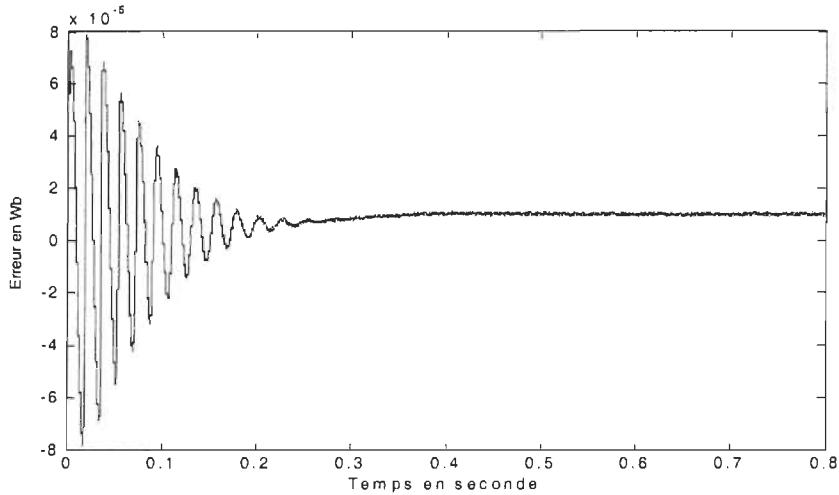


Figure 1.6 : L'erreur d'estimation du flux estimé par rapport au flux réel

Notons que d'une part le flux estimé varie de façon semblable au flux réel presque partout, dans le régime permanent, et que d'autre part l'erreur de variation convergent vers zéro.

1.3.3. L'observateur de Luenberger

1.3.3.1. Principes

Il est essentiel de faire un petit rappel sur quelques principes fondamentaux de cet observateur.

Soit le système linéaire, d'entrée u , de sortie y et d'état x , décrit par :

$$\begin{cases} \dot{X} = AX + BU \\ Y = CX + DU \end{cases} \text{ avec } X(t) \in \mathbb{R}^n, U(t) \in \mathbb{R}^m, Y(t) \in \mathbb{R}^p \quad (1.6)$$

Si la paire (C, A) est observable alors le système, décrit par (1.8), est un observateur de (1.6) si et seulement si :

$$\lim_{t \rightarrow \infty} (X(t) - \hat{X}(t)) \rightarrow 0 \quad (1.7)$$

$$\begin{cases} \dot{\hat{X}} = A\hat{X} + BU + G(Y - \hat{Y}) \\ \hat{Y} = C\hat{X} + DU \end{cases} \text{ avec } \hat{X}(t) \in \mathbb{R}^n, U(t) \in \mathbb{R}^m, \hat{Y}(t) \in \mathbb{R}^p \quad (1.8)$$

Nous constatons qu'il est facilement démontrable que la condition (1.5) est vérifiée si la matrice $(A-GC)$ est une matrice de stabilité car, elle possède des valeurs propres à parties réelles strictement négatives. C'est parce que les observateurs suivants sont basés sur ce principe qu'ils sont qualifiés de type Luenberger.

On ne peut pas appliquer ce principe d'une façon générale et directe à cause de la non linéarité du modèle de la machine asynchrone, mais selon le type de modèle utilisé, le référentiel choisi et sous certaines hypothèses, ce serait possible de l'appliquer dans plusieurs types d'observateurs basés sur ce principe. Cependant, nous présenterons dans cette partie seulement ceux qui sont les plus utilisés.

1.3.3.2. L'observateur de Verghese

Il possède une structure simple et à convergence quadratique. Quand sa construction est qualifiée de construction simple, c'est pour dire qu'il est calculé sur la base d'un observateur de Luenberger et qu'il est de dimension raisonnable. Ensuite, il est aussi de structure linéaire à paramètres variants.

Pour que notre observateur possède le même niveau de convergence et de performance que l'observateur de Luenberger, Verghese part de la constatation suivante: à vitesse mécanique constante le sous-système électrique de la machine asynchrone (2.2.5 équation 3.22) est linéaire. On propose donc de construire un observateur de Luenberger sur ce sous-système et de faire une correction (ou injection de sortie) dépendante de la vitesse du rotor. Donc, en effectuant la séparation du sous-modèle électrique de la machine et du sous modèle mécanique dans le référentiel (α, β) , le système est linéaire à paramètres dépendants de la vitesse du rotor au [2]:

L'équation d'état est donnée par :

$$\begin{cases} \dot{X} = A(\Omega)X + BU \\ Y = CX \end{cases}$$

$$\text{Soit : } \frac{d}{dt} \begin{bmatrix} \Phi_{r\alpha} \\ \Phi_{r\beta} \\ i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{L_r} & -p\Omega & \frac{R_r}{L_r} M_s & 0 \\ p\Omega & \frac{-R_r}{L_r} & 0 & \frac{R_r}{L_r} M_s \\ \frac{M_s R_r}{\sigma L_s L_r} & \frac{p M_s \Omega}{\sigma L_s L_r} & -\gamma_m & 0 \\ \frac{-p M_s \Omega}{\sigma L_s L_r} & \frac{M_s R_r}{\sigma L_s L_r} & 0 & -\gamma_m \end{bmatrix} \begin{bmatrix} \Phi_{r\alpha} \\ \Phi_{r\beta} \\ i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \end{bmatrix} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad \text{et} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (1.9)$$

$$\text{Avec : } \sigma = 1 - \frac{M_s^2}{L_s L_r}, \gamma_m = \frac{L_r^2 R_s + M_s^2 R_r}{\sigma L_s L_r}$$

$$\text{Et} \quad U = \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad Y = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad X(t) = \begin{bmatrix} \Phi_{r\alpha} \\ \Phi_{r\beta} \\ i_{s\alpha} \\ i_{s\beta} \end{bmatrix}$$

L'observateur de Verghese est :

$$\begin{cases} \dot{\hat{X}} = A(\omega_m) \hat{X} + BU + G(Y - \hat{Y}) \\ \hat{Y} = C\hat{X} + DU \end{cases} \quad (1.10)$$

$$\text{avec : } \hat{X} = \begin{bmatrix} \hat{\Phi}_{r\alpha} \\ \hat{\Phi}_{r\beta} \\ \hat{i}_{s\alpha} \\ \hat{i}_{s\beta} \end{bmatrix}, \quad U = \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix}, \quad Y = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}$$

L'observateur de Verghese possède une structure du gain de l'injection de sortie particulière (1.11) car, il présente certaines symétries et est dépendant de la vitesse du rotor.

$$G = \begin{bmatrix} g_1 & -g_2 \omega \\ g_2 \omega & g_1 \\ g_3 & -g_4 \omega \\ g_4 \omega & g_3 \end{bmatrix} \quad (1.11)$$

Sous la condition qu'il n'y ait pas de polarisation sur les paramètres du processus et sur les grandeurs mesurées, l'erreur d'estimation $e = X - \hat{X}$ satisfait l'équation:

$$\dot{e} = (A(\omega) - G(\omega)C)e = \tilde{A}(\omega)e \quad (1.12)$$

L'expression (1.12) assure une bonne convergence exponentielle pour l'observateur de Verghese. Pour satisfaire la stabilité de l'observateur, on choisit la matrice de gain G de façon que les pôles de $\tilde{A}(\omega)$ soient à parties réelles strictement négatives. La grandeur $\tilde{A}(\omega)$ déterminant la dynamique de l'erreur d'observation s'écrit alors sous la forme donnée par l'équation (1.13).

$$\tilde{A}(\omega) = \begin{bmatrix} \frac{-R_r}{L_r} & 0 & \frac{M_{sr}R_r}{L_r} & 0 \\ 0 & \frac{-R_r}{L_r} & 0 & \frac{M_{sr}R_r}{L_r} \\ \frac{M_{sr}R_r}{d^2 L_s} & 0 & \frac{M^2_{sr}R_r + R_s L_r^2}{d^2 L_s} & 0 \\ 0 & \frac{M_{sr}R_r}{d^2 L_s} & 0 & \frac{M^2_{sr}R_r + R_s L_r^2}{d^2 L_s} \end{bmatrix} + p\Omega \begin{bmatrix} 0 & -1 & 0 & -g_2 \\ 1 & 0 & g_2 & 0 \\ 0 & \frac{M_{sr}}{d_s L_r} & 0 & -g_4 \\ \frac{-M_{sr}}{d_s L_r} & 0 & g_4 & 0 \end{bmatrix} \quad (1.13)$$

On peut déterminer la dynamique de l'erreur d'observation par un placement de pôles. En considérant que les dynamiques des flux et des courants ont des constantes de temps voisines. Verghese nous propose un choix de gains tels que g_1, g_2, g_3 et g_4 pour simplifier (1.13) de manière significative et obtenir (1.14).

$$g_1 = -\frac{M_{sr}R_r}{L_r} - \frac{R_r}{L_r}g_2 \text{ et } g_3 = \gamma_m - \frac{R_r}{L_r}g_4 = \frac{L_r^2 R_s + M_{sr}^2 R_r}{\sigma L_s L_r^2} - \frac{R_r}{L_r}g_4$$

$$\tilde{A}(\Omega) = B \cdot Q(\Omega) = \begin{bmatrix} 1 & 0 & g_2 & 0 \\ 0 & 1 & 0 & g_2 \\ -\frac{M_{sr}R_r}{L_r} & 0 & g_4 & 0 \\ 0 & -\frac{M_{sr}R_r}{L_r} & 0 & g_4 \end{bmatrix} \begin{bmatrix} -\frac{R_r}{L_r} & -p\Omega & 0 & 0 \\ -p\Omega & -\frac{R_r}{L_r} & 0 & 0 \\ 0 & 0 & -\frac{R_r}{L_r} & -\Omega \\ 0 & 0 & -\Omega & -\frac{R_r}{L_r} \end{bmatrix} \quad (1.14)$$

Cette simplification permet un calcul ais   des modes propres de l'erreur d'estimation. En effet, il est possible de calculer s  par  m  t les valeurs propres des deux matrices B et $Q(\Omega)$, cette derni  re   tant diagonale par blocs. Finalement, les gains de l'observateur sont d  duits en proc  dant de la mani  re suivante:

- les valeurs propres doubles λ_1 et λ_2 de la matrice B sont les solutions de (1.15)

$$[\lambda^2 - (1+g_4)\lambda + g_4 + \frac{M_{sr}}{\sigma L_s L_r} g_2]^2 = 0 \quad (1.15)$$

- les valeurs propres de la matrice $\tilde{A}(\Omega)$ sont:

$$p_{1,2} = \left[\left(\frac{-1}{\tau} + -jp\Omega \right) \right] \lambda_1 \text{ et } p_{3,4} = \left[\left(\frac{-1}{\tau} + -jp\Omega \right) \right] \lambda_2 \quad (1.16)$$

1.3.3.3. L'observateur de Kalman

1. M  thode d  terministe : l'observateur d  terministe se base essentiellement sur les   quations des syst  mes sans prendre en compte des bruits de mesure ou les perturbations. Cet observateur se caract  rise donc la plupart du temps par une sensibilit   aux perturbations et aux variations param  triques.
2. M  thode stochastique : le filtre de Kalman prend en consid  ration les bruits de mesure et d'  tat par des algorithmes stochastiques qui tend    minimiser la variance de l'erreur d'estimation. En effet, l'algorithme du filtre requiert un nombre de calculs important induit par la pr  sence d'inverse de matrices. En phase d'impl  mentation, ce nombre important de calculs est p  nalisant en pratique car il limite alors le nombre d'op  rations n  cessaires    la commande proprement dite sauf si la commande est r  alis  e    l'aide de DSP performant et efficace.

Un observateur de Kalman, qu'il soit d  terministe ou stochastique, lin  aire ou non lin  aire, travaille g  n  eralement en deux temps (voir figure 1.6) :

1.   tage de pr  diction : il effectue une pr  diction $\hat{X}[k+1/k]$ de l'  tat du processus    partir d'un mod  le d'  tat con  u    partir des   quations du syst  me. Cette   tage est g  n  eralement assez pr  cise.

2. Étape de correction : il effectue la correction de la valeur des variables d'état issues de l'étape précédente pour obtenir $\hat{X}[k+1/k+1]$ à partir des mesures effectuées sur le processus réel, et corrige ainsi les résultats de la partie prédictive.

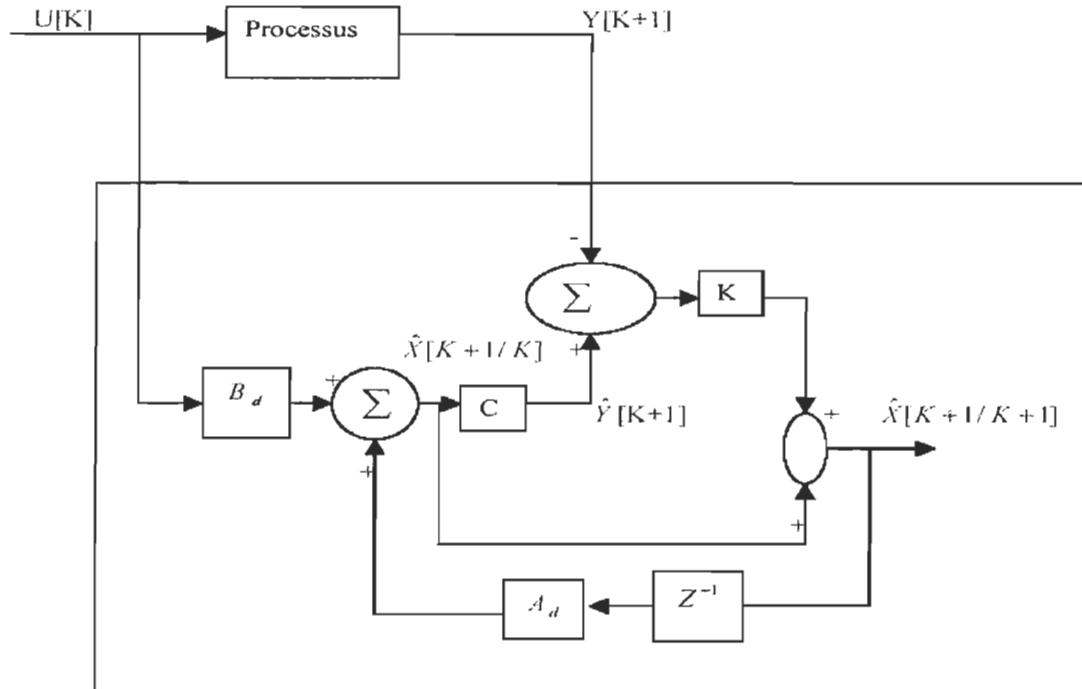


Figure1.7 :Schéma bloc d'un filtre de Kalman

Comme la sortie de l'observateur de Kalman est asservie à la sortie réelle du processus, les observateurs sont équivalents à des asservissements et donc à des filtres passe-bas. Ainsi dans le cadre stochastique, le terme de filtre est préféré à celui d'observateur d'état plus communément usité dans le cadre de l'estimation déterministe bien que les structures soient totalement identiques.

1.3.3.5. Observateur par modes glissants

Cet observateur est basé sur les méthodes employées dans la commande par modes glissants [7] et dans la pratique, il s'avère le plus performant et le plus robuste. Jusqu'à maintenant, il n'existe pas de démonstration rigoureuse permettant de prouver ses

qualités. Dans la paragraphe qui suit, une méthode de synthèse qui s'avère être une des plus rigoureuse [7] est exposée.

Cette partie a pour objectif principal d'expliquer la méthode de synthèse de ce type d'observateur qui est encore appelé aussi observateur à structure variable. L'observateur par modes glissants donné par (1.17) estime les flux et les courants à partir des courants statoriques et de la vitesse mécanique [5]. Sa structure est identique à celle de l'observateur proposé par Verghese; seule la forme de l'injection de sortie change.

$$\begin{cases} \dot{\hat{X}} = A(\Omega) \hat{X} + BU + G \cdot \text{signe} (Y - \hat{Y}) \\ \hat{Y} = C\hat{X} \end{cases} \quad (1.17)$$

$$\text{avec : } \hat{X} = \begin{bmatrix} \hat{\Phi}_{r\alpha} \\ \hat{\Phi}_{r\beta} \\ \hat{i}_{s\alpha} \\ \hat{i}_{s\beta} \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \hat{i}_{s\alpha} \\ \hat{i}_{s\beta} \end{bmatrix}$$

La méthode de synthèse, proposée dans [5], se base essentiellement sur la théorie des modes glissants. Cette méthode se résume aux points présentés ci-dessous :

- Poser les erreurs de reconstruction (1.18) comme surfaces et écrire les conditions de stabilité des surfaces considérées.

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} = \begin{bmatrix} \phi_{r\alpha} - \hat{\phi}_{r\alpha} \\ \phi_{r\beta} - \hat{\phi}_{r\beta} \\ i_{s\alpha} - \hat{i}_{s\alpha} \\ i_{s\beta} - \hat{i}_{s\beta} \end{bmatrix} \quad (1.18)$$

- Écrire les conditions de stabilité des surfaces considérées :

$$\begin{cases} e_i \dot{e}_i \prec 0 \\ \forall i \in N, i \in [1;4] \end{cases} \quad (1.19)$$

La condition de stabilité de Lyapunov est alors : $\dot{V}(x) \prec 0 \Leftrightarrow e_i \dot{e}_i \prec 0$ (1.20)

- Choisir pour l'injection de sortie une matrice de gain de la forme :

$$G = \begin{bmatrix} -g_1 & -g_2 \\ -g_2 & -g_1 \\ -K_1 & K_2 \\ -K_2 & -K_1 \end{bmatrix} \quad (1.21)$$

- À partir de (1.20), les conditions de stabilité de Lyapunov des surfaces glissantes de la structure de la matrice de gain G ainsi que des limites de fonctionnement de la machine Sangwongwanich et al. [5] déterminent les gains de commutation par minimisation d'un critère H_∞ sur l'erreur de reconstruction.

Discussion :

Il apparaît que l'observateur à modes glissants proposé par [5], répond à certains critères de robustesse, prouvés par son utilisation [8], du fait qu'il est très peu sensible aux variations de modèle. Cette qualité est très importante, mais à cause de la forme particulière de son injection de sortie (fonction signe) qui engendre des oscillations hautes fréquences en régime permanent, il n'est pas très utilisé contrairement aux autres types d'observateurs.

1.3.3.6. L'observateur à grands gains

L'observateur à grands gains est basé sur la théorie de l'étude des systèmes non-linéaires (1.22).

$$\begin{cases} \dot{X} = f(X, U) \\ Y = h(X, U) \end{cases} \quad \text{où } X(t) \in \mathbb{R}^n, U(t) \in \mathbb{R}^m, Y(t) \in \mathbb{R}^p \quad (1.22)$$

L'idée de base est de chercher une transformation d'espace du type $\xi = \Phi(x)$, tel que le système (1.22) puisse s'écrire sous la forme canonique d'observabilité suivante :

$$\begin{cases} \dot{\xi} = A\xi + \varphi(\xi, U) \\ Y = C\xi \end{cases} \quad \text{où } \xi \in \mathbb{R}^n \quad (1.23)$$

- Un système non linéaire[4], (Σ) décrit par (1.22) admet une forme canonique d'observabilité (1.24) si et seulement si:

1. Le système (Σ) est observable,
2. Il existe localement p entiers tels que:

$$(\xi_1, \dots, \xi_n) = (h_1, L_f h_1, \dots, L_f^{n_1} h_1, \dots, h_p, \dots, L_f^{n_p} h_p) \quad (1.24)$$

$$\text{avec } \sum_{i=1}^p \eta_i = n$$

Ces deux conditions vérifiées, l'observateur à grands gains s'écrit par:

$$\begin{cases} \dot{\hat{\xi}} = A\hat{\xi} + \varphi(\xi, U) + \Delta^{-1}K(y - \hat{y}) \\ y = C\xi \end{cases} \quad (1.25)$$

avec pour $k=1$ à p

$$\begin{cases} A = \text{diag}(A_1, \dots, A_p) \\ C = \text{diag}(C_1, \dots, C_p) \end{cases}, \quad A_k = \begin{bmatrix} 0 & 1 & \dots & \dots & 0 \\ 0 & 0 & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad C_K = [1, 0, \dots, 0] \quad (1.26)$$

$$\Delta = \text{diag}(\Delta_1, \dots, \Delta_p), \quad \Delta_K = \text{diag}(T^{\delta_1}, \dots, T^{\delta_p}),$$

1. les indices δ_k sont calculés de manière algorithmique suivant la méthode exposée dans [4],
2. La matrice de gain, $K = \text{diag}(K_1, \dots, K_p)$ est choisie telle que pour chaque bloc K_i , la matrice $(A_i - K_i C_i)$ soit une matrice stable,

3. T est choisie relativement faible.

Un observateur peut être construit à partir du modèle (1.7) de la machine asynchrone sans la position mécanique du rotor dans l'espace d'état .

Les mesures disponibles sont : $y = [\Omega, i_{s\alpha}, i_{s\beta}]' = h(x)$ qui vont nous permettre alors de déterminer le changement de base suivant : $(\xi_1, \dots, \xi_5) = (h_2, L_f h_2, h_3, L_f h_3, h_1)$

Soit :

$$\left\{ \begin{array}{l} \xi_1 = i_{s\alpha} \\ \xi_2 = -\gamma i_{s\alpha} + \frac{\alpha M_{sr}}{\sigma L_s L_r} \phi_{ra} + \frac{\alpha M_{sr}}{\sigma L_s L_r} p \Omega \phi_{r\beta} \\ \xi_3 = i_{s\beta} \\ \xi_4 = -\gamma i_{s\beta} + \frac{\alpha M_{sr}}{\sigma L_s L_r} \phi_{r\beta} - \frac{\alpha M_{sr}}{\sigma L_s L_r} p \Omega \phi_{r\alpha} \\ \xi_5 = \Omega \end{array} \right. \quad (1.27)$$

Ceci permet d'écrire l'observateur à grands gains pour la machine asynchrone.

$$\left\{ \begin{array}{l} \dot{\xi} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{V_{s\alpha}}{\sigma L_s} \\ \varphi_2(\xi, U) \\ \frac{V_{s\beta}}{\sigma L_s} \\ \varphi_4(\xi, U) \\ \varphi_5(\xi, U) \end{bmatrix} + \Delta^{-1} K \begin{bmatrix} i_{s\alpha} - \hat{i}_{s\alpha} \\ i_{s\beta} - \hat{i}_{s\beta} \\ \Omega - \hat{\Omega} \end{bmatrix} \\ \hat{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \xi \end{array} \right. \quad (1.28)$$

Avec les coefficients simplificateurs (1.29)

$$\left\{ \begin{array}{l} F_2 = \frac{M_{sr}}{\sigma L_s L_r} \left(\frac{R_r}{L_r} (\xi_2 + \gamma_m \xi_1) - p \xi_5 (\xi_4 + \gamma_m \xi_3) \right) \\ F_3 = \frac{M_{sr}}{\sigma L_s L_r} \left(\frac{R_r}{L_r} (\xi_4 + \gamma_m \xi_3) - p \xi_5 (\xi_2 + \gamma_m \xi_1) \right) \\ A = \frac{M_{sr}^2}{\sigma L_s L_r} \left(\left(\frac{R_r}{L_r} \right)^2 + p^2 \xi_5^2 \right) \end{array} \right. \quad (1.29)$$

Et enfin, la matrice de gain fixant la dynamique de l'observateur est donnée par (1.31).

$$\left\{ \begin{array}{l} \varphi_2 = -\gamma(\xi_2 + \frac{V_{s\alpha}}{\sigma L_s}) + \frac{\alpha M_{sr}}{\sigma L_s L_r} \left[\frac{-\alpha F_2 + p F_3 \xi_5}{A} + \alpha M_{sr} \xi_1 - \frac{p}{\alpha} \left(\frac{\dot{\Omega} F_2}{A} + \xi_5 \left(\frac{-\alpha F_2 - p F_3 \xi_5}{A} + \alpha M_{sr} \xi_1 \right) \right) \right] \\ \varphi_4 = -\gamma(\xi_4 + \frac{V_{s\beta}}{\sigma L_s}) + \frac{\alpha M_{sr}}{\sigma L_s L_r} \left[\frac{-\alpha F_3 + p F_2 \xi_5}{A} + \alpha M_{sr} \xi_3 - \frac{p}{\alpha} \left(\frac{\dot{\Omega} F_3}{A} + \xi_5 \left(\frac{-\alpha F_2 - p F_3 \xi_5}{A} + \alpha M_{sr} \xi_3 \right) \right) \right] \\ \varphi_5 = \dot{\Omega} \end{array} \right. \quad (1.30)$$

$$K = \begin{bmatrix} K_1 & 0 & 0 \\ K_2 & 0 & 0 \\ 0 & K_3 & 0 \\ 0 & K_4 & 0 \\ 0 & 0 & K_5 \end{bmatrix} \quad (1.31)$$

L'algorithme disponible dans [4] permet de calculer tous les coefficients σ et δ pour en déduire la matrice Δ :

$$\Delta = \text{diag}(T, T^2, T, T^2, T) \quad (1.32)$$

Discussion :

Un observateur à grands gains n'est pas facile à déterminer comme le montre les équations précédentes. De plus, la programmation de ce type d'observateur se fait en recalculant le système (1.28) dans la base originale. Ceci, montre que l'existence d'un observateur à grands gains dépend de l'existence de la transformation inverse, cette dernière dépendant du choix des variables ξ_i . Notons de plus l'importance d'avoir une mesure du couple de charge car celui-ci intervient directement sur la dernière composante de l'état observé. C'est l'inconvénient d'un observateur à grands gains d'ordre complet.

1.4. L'observateur de flux linéaire à paramètre variant (LPV) :

Le modèle le plus utilisé pour la synthèse d'observateurs de flux, est donné par le sous-modèle électrique de la machine. Ce modèle (1.33) est composé des flux rotoriques et des courants statoriques exprimés dans un repère diphasé fixe par rapport au stator.

$$\frac{d}{dt} \begin{bmatrix} \Phi_{r\alpha} \\ \Phi_{r\beta} \\ i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{L_r} & -p\Omega & \frac{R_r M_s}{L_r} & 0 \\ p\Omega & -\frac{R_r}{L_r} & 0 & \frac{R_r M_s}{L_r} \\ \frac{M_s R_r}{\sigma L_s^2} & \frac{p M_s \Omega}{\sigma L_s} & -\gamma_m & 0 \\ -\frac{p M_s \Omega}{\sigma L_s} & \frac{M_s R_r}{\sigma L_s^2} & 0 & -\gamma_m \end{bmatrix} \begin{bmatrix} \Phi_{r\alpha} \\ \Phi_{r\beta} \\ i_{s\alpha} \\ i_{s\beta} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \end{bmatrix} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad \text{et} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (1.33)$$

L'expression (1.24) montre que ce modèle est linéaire à un paramètre près. Ce paramètre est la vitesse mécanique qui possède l'avantage d'évoluer plus lentement que les paramètres électriques et de surcroît dans un ensemble borné. Cet ensemble est naturellement défini par le régime nominal de fonctionnement de la MAS. Cette méthode d'observation quant à elle se distingue par sa robustesse, cependant son coût algorithmique est moins performant par rapport à la méthode stochastique (le filtre de Kalman optimal) (Référence : Annexe2).

1.5. Synthèse d'un observateur LMI linéaire à paramètres variants

L'observateur LPV permet de fournir une estimation du flux rotorique quel que soit le régime de fonctionnement à partir des mesures des courants statoriques et de la vitesse du rotor. On considère le domaine de fonctionnement (1.34), qui a été choisi plus vaste que le domaine de fonctionnement normal de la machine, afin que l'observateur à synthétiser puisse répondre éventuellement à des régimes de survitesse [2,8].

$$\Omega \in [-2\Omega_{\text{nom}}, 2\Omega_{\text{nom}}] \quad (1.34)$$

De plus dans le cadre d'une synthèse L_2 d'un observateur LPV, la vitesse du rotor est supposée évoluer dans un ensemble polytopique:

$$S_\Omega = \text{Co}\{\underline{\Omega}, \bar{\Omega}\} = \text{Co}\{-2\Omega_{\text{nom}} ; 2\Omega_{\text{nom}}\}, \forall \Omega \in S_\Omega, \exists \alpha \in [0,1] \\ / \Omega = \alpha \underline{\Omega} + (1-\alpha)\bar{\Omega} \quad (1.35)$$

L'objectif dans cette partie est de trouver un observateur de flux quadratiquement stable, assurant une atténuation L_2 spécifique entre les erreurs de reconstruction et d'éventuelles perturbations.

1.5.1. Problème standard associé à un observateur de flux LPV

La description LPV est paramétré par la vitesse du rotor auquel nous associons les sorties relatives aux grandeurs d'états observées (y_m) ainsi que celles relatives aux grandeurs d'états à reconstruire (e_o).

D'après le cahier des charges ces grandeurs sont :

$$e_o = \begin{pmatrix} \phi_{r\alpha} \\ \phi_{r\beta} \end{pmatrix}, y_m = \begin{pmatrix} i_{s\alpha} \\ i_{s\beta} \end{pmatrix} \quad (1.36)$$

Les mesures des courants statoriques que voici sont supposées corrompues par des bruits (w_1). Ces bruits ne sont supposés agir que dans une certaine bande de fréquence. C'est pourquoi leur influence est caractérisée via une pondération (w). De même, les tensions statoriques de commande sont supposées corrompues par une perturbation (w_2) afin de tenir compte dans le critère des non linéarités de l'onduleur. Ceci permet de traduire le fait que les tensions calculées par l'algorithme de commande ne sont pas celles réellement appliquées à la machine. Dans ce cas, le vecteur de mesure considéré pour concevoir l'observateur est composé des courants statoriques bruités et des tensions de commandes ($u_{s\alpha\beta}$) fournies par le régulateur.

Enfin, il est défini par (\hat{e}_o) l'estimation du flux et par e_o l'erreur d'estimation (1.37).

$$\left\{ \begin{array}{l} e_o = \phi_{r\alpha,\beta} \\ \hat{e}_o = \hat{\phi}_{r\alpha,\beta} \\ e_o = e_o - \hat{e}_o = \phi_{r\alpha,\beta} - \hat{\phi}_{r\alpha,\beta} \end{array} \right. \quad (1.37)$$

finalement, le problème standard associé à l'observateur de flux LPV est décrit par le schéma de la figure (1.7).

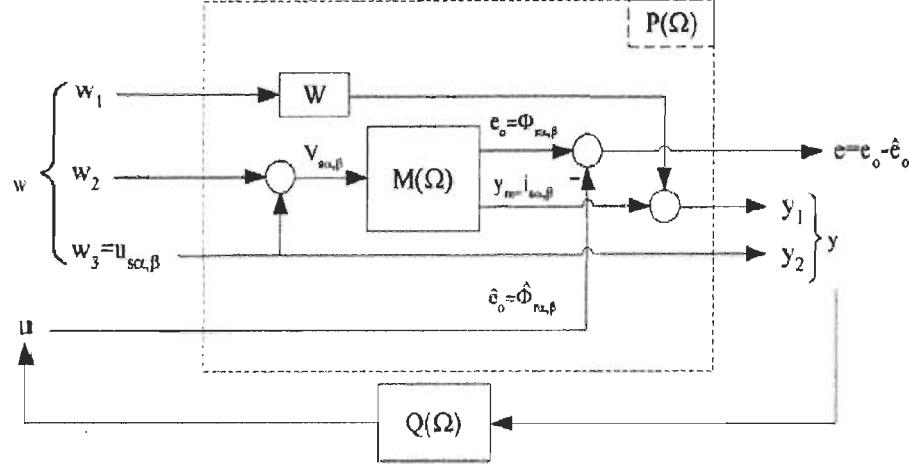


Figure 1.8: Problème standard associé à la synthèse d'un observateur de flux LPV pour la MAS

Nous pouvons démontrer que la matrice système (1.38) du problème standard est affine en la vitesse du rotor si on suppose la pondération (w) indépendante de celle-ci et en utilisant la propriété d'interconnexion des LFT.

$$P(\Omega) = \begin{bmatrix} A_{pc,1} & B_{pc,1} & B_{pc,12} \\ C_{pc,1} & D_{pc,11} & D_{pc,12} \\ C_{pc,2} & D_{pc,21} & D_{pc,22} \end{bmatrix} + \Omega \begin{bmatrix} A_{pv,1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1.38)$$

Notre problème standard ainsi défini est linéaire à paramètres variants LPV puisqu'il dépend linéairement de la vitesse du rotor (1.38). En plus, ce paramètre évolue dans un ensemble borné. Un régulateur LPV répondant à l'ensemble des objectifs de l'observation peut être synthétisé en solutionnant le problème L_2 suivant: le problème Q-H ∞ optimal d'observation du flux rotorique d'une MAS est de synthétiser un observateur $Q(\Omega)$ tel que :

$$\left\{ \begin{array}{l} J(Q(\Omega)) = \|F_1(P(\Omega), Q(\Omega))\|_{\infty} \stackrel{\Delta}{=} \sup \frac{\|e\|_2}{\|w\|_2} \leq \gamma \\ \gamma \leq 1 \end{array} \right. \quad (1.39)$$

sous contrainte de stabilité quadratique de la boucle fermée (figure1.7) et, pour l'ensemble des trajectoires admissibles de la vitesse du rotor (Ω) (1.39) :

Le critère adopté n'est pas unique et peut être amélioré pour tenir compte des variations de certains paramètres tels les résistances statoriques et /ou rotoriques. La configuration actuelle employée possède un seul degré de liberté induit par la pondération (w). Ceci permet d'obtenir un modèle standard $P(\Omega)$ de dimension raisonnable et par conséquent un observateur $Q(\Omega)$ aussi de dimension raisonnable car de même ordre.

Il est important de prendre en considération que le modèle standard n'est pas commandable par (u). L'existence d'une solution stabilisante au problème est donc assujettie par la stabilité du modèle standard.

1.5.2. Réglage de la bande passante de l'observateur

Notre objectif est de régler la bande passante de l'observateur à synthétiser, la fonction (w) permet de pondérer les deux mesures de courants (i_{sa} et i_{sb}). La bande passante de l'observateur est inchangée, quel que soit le régime de fonctionnement du moteur. La pondération est choisie indépendante du paramètre variant de la vitesse de rotation du rotor. Afin de simplifier l'ordre (pas l'augmenter inutilement) du modèle standard et, par conséquent l'ordre de la solution, la pondération proposée est du premier ordre et ne possède qu'un minimum de degré de liberté. Enfin, elle doit permettre une forte atténuation dans la bande passante désirée et garantir un certain niveau au-delà.

La matrice de transfert $W(s)$ choisie pour répondre à l'ensemble des contraintes est :

$$(W): = W(s) = \begin{bmatrix} \frac{g_b s}{s + w_b} & 0 \\ 0 & \frac{g_b s}{s + w_b} \end{bmatrix} = W(s).I_2 \quad (1.40)$$

1.5.3. Synthèse et analyse de l'observateur de flux LPV

En supposant que la vitesse de rotation du rotor est mesurée en temps réel. Le problème précédent (1.39) peut être écrit sous la forme d'un problème d'optimisation convexe sous

contraintes LMI, en considérant une description polytopique du problème standard $P(\Omega)$. L'observateur LPV est finalement obtenu en utilisant la boîte à outil *LMI-control toolbox* de ©Matlab [25]. Cet observateur est donné sous forme polytopique pour lequel il en est facilement déduit la forme affine (LPV) (1.41). De plus, il est de même ordre et de même structure que le modèle $P(\Omega)$ du système standard. Dans le cas présent, l'observateur de flux synthétisé est du sixième ordre (no=6).

$$\begin{cases} \begin{bmatrix} x_0 \\ \hat{\phi}_{s\alpha,\beta} \end{bmatrix} = \begin{bmatrix} A_{02} + \alpha(A_{01} - A_{02}) & B_{02} + \alpha(B_{01} - B_{02}) \\ C_{02} + \alpha(C_{01} - C_{02}) & D_{02} + \alpha(D_{01} - D_{02}) \end{bmatrix} \begin{bmatrix} x_0 \\ I_{s\alpha,\beta} \\ V_{s\alpha,\beta} \end{bmatrix} \\ \alpha = \frac{\bar{\Omega} - \Omega}{\Omega - \bar{\Omega}} \end{cases}$$

$x_0(t) \in \mathbb{R}^6, i_{s\alpha,\beta}(t) \in \mathbb{R}^2, V_{s\alpha,\beta}(t) \in \mathbb{R}^2, \hat{\phi}_{s\alpha,\beta}(t) \in \mathbb{R}^2 \text{ et } \alpha(t) \in \mathbb{R}$ (1.41)

avec $S_0 = \begin{bmatrix} A_{01} & B_{01} \\ C_{01} & D_{01} \end{bmatrix}$ et $\bar{S}_0 = \begin{bmatrix} A_{02} & B_{02} \\ C_{02} & D_{02} \end{bmatrix}$ sont les deux sommets de sa représentation polytopique.

1.5.3.1. Méthodologie de discréétisation de l'observateur

1.5.3.1.1. Introduction

Dans ce paragraphe nous allons traiter les problèmes de discréétisation de l'observateur proposé et les méthodes de résolution misent en jeu. Pour les systèmes LTI, il existe plusieurs méthodes permettant l'obtention d'un système discret équivalent, qu'on peut ranger en deux classes [20]:

1. La première regroupe un ensemble de méthodes dont l'objectif est de discréétiser la réponse du système régie de la solution générale de l'équation différentielle décrivant le système.
2. Le second type de méthodes sont des méthodes de substitution qui consistent à remplacer l'opérateur d'intégration de la description LFT du système à discréétiser.

Si on applique le premier type de méthodes à notre système LPV et, en particulier à l'observateur de flux, nous remarquons que même en supposant constants les signaux d'entrée de l'observateur $i_{s\alpha,\beta(t)}, v_{s\alpha,\beta(t)}$ et $\Omega(t)$ pendant une période d'échantillonnage. la simulation numérique de l'observateur nécessite le calcul en ligne d'inverse et d'exponentielle de matrices (1.42) ce qui exige un important coût de calcul, au profil des autres opérations donc cette méthode est jugée inutile.

$$x_{0,k+1} = e^{A_0(\alpha_k)T_e} x_{0,k} + (A_0(\alpha_k))^{-1} (e^{A_0(\alpha_k)T_e} - I) B_0(\alpha_k) \begin{bmatrix} i_{s\alpha,\beta,K} \\ v_{s\alpha,\beta,K} \end{bmatrix}$$

avec $\begin{cases} A_0(\alpha_k) = A_{02} + \alpha_k + \alpha_k(A_{01} - A_{02}) \\ B_0(\alpha_k) = B_{02} + \alpha_k + \alpha_k(B_{01} - B_{02}) \end{cases}$ (1.42)

Si on applique à l'observateur la seconde classe de méthode qui consiste à intégrer (5.9), alors on note une très nette diminution du coût de calcul (exemple: si l'on utilise une intégration de type Euler). Ainsi la précision n'est pas vraiment intéressante lors de la reconstruction que pour de très petites périodes d'échantillonnage incompatibles avec le matériel disponible. Pour améliorer, ceci, Apkarian adopte dans [21] l'utilisation de la méthode des trapèzes ou de Tustin pour résoudre (1.41) qui présente de sa part un inconvénient qui est l'introduction d'une boucle algébrique qu'il faut résoudre pour obtenir le système discret. En contre partie, cette résolution requiert l'inversion en ligne d'une matrice fonction du paramètre externe Ω [21].

1.5.3.1.2. Méthode proposée

La méthode que nous proposons pour discréteriser l'observateur de flux se classe parmi le deuxième type de méthode dite de substitution. L'idée de base consiste à la substitution par un approximant discret de l'opérateur d'intégration (I_c) de la description LFT de l'observateur (figure 1.9).

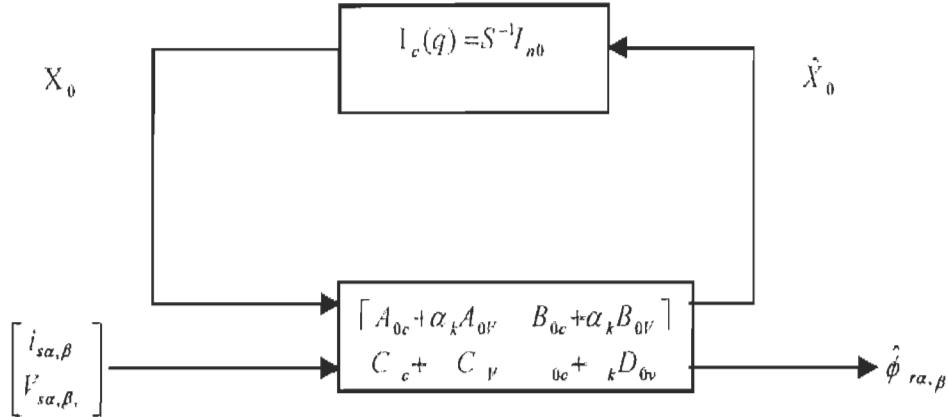


Figure 1.9 : Description LFT de l'observateur LPV

L'intégrant discret obtenu ne devra pas de toute façon introduire de bouclage algébrique. Par conséquent, l'opérateur intégral continu $I_c(s) = s^{-1} I_{n0}$ est substitué par un équivalent discret $I_d(q^{-1})$ qui devra impérativement être strictement propre. De plus, il est important de noter que tous les signaux internes de l'observateur sont de nature sinusoïdale et de fréquence proche de celle des tensions statoriques. C'est pourquoi, il peut être intéressant d'avoir un opérateur qui laisse ce type de signal invariant [20]. Dans ce contexte, il reste alors deux possibilités: soit l'opérateur laisse le sinus invariant, soit il laisse le cosinus invariant. L'obtention d'un opérateur d'intégration laissant un type particulier de signal invariant s'obtient par le ratio de la transformée en (z) du signal intégré et de la transformée en (z) du signal à intégrer [21].

Remarque : Rappelons que q est l'opérateur avance et S est l'opérateur de dérivation.

- L'opérateur d'intégration laissant le sinus invariant est donné par :

$$I_{d1}(q) = \frac{\omega^{-1} (\cos(\omega T_e) - q)}{\sin(\omega T_e)} \quad (1.43)$$

- Démonstration : Le signal sinusoïdal à intégrer est décrit par (1.43) et sa primitive par (1.44).

$$u(t) = \sin(\omega t) \xrightarrow{z} u_i(z) = \frac{z^{-1} \sin(\omega T_e)}{z^{-2} - 2z^{-1} \cos(\omega T_e) + 1} \quad (1.44)$$

$$u_i(t) = \omega^{-1} \cos(\omega t) \xrightarrow{z} u_i(z) = \omega^{-1} \frac{(1 - z^{-1} \cos(\omega T_e))}{z^{-2} - 2z^{-1} \cos(\omega T_e) + 1} \quad (1.45)$$

L'intégrateur discret laissant le sinus invariant est alors obtenu suivant (1.46).

$$I_{d1}(Z) = \frac{u_i(z)}{u(z)} = -\omega^{-1} \frac{(1-z^{-1} \cos(\omega T_e))}{z^{-1} \sin(\omega T_e)} \quad (1.46)$$

L'opérateur défini par (1.43) n'étant pas strictement propre il n'est pas retenu.

- L'opérateur d'intégration discret laissant le cosinus invariant est donné par :

$$I_{d2}(q) = \omega^{-1} \frac{\sin(\omega T_e)}{\omega(q - \cos(\omega T_e))} \quad (1.47)$$

- Démonstration : Le signal sinusoïdal à intégrer est décrit par (1.47) et sa primitive par (1.48).

$$u(t) = \cos(\omega t) \xrightarrow{z} u(z) = \frac{(1-z^{-1} \cos(\omega T_e))}{z^{-2} - 2z^{-1} \cos(\omega T_e) + 1} \quad (1.48)$$

$$u_i(z) = \omega^{-1} \sin(\omega t) \xrightarrow{z} u(z) = \omega^{-1} \frac{z^{-1} \sin(\omega T_e)}{z^{-2} - 2z^{-1} \cos(\omega T_e) + 1} \quad (1.50)$$

L'intégrateur discret laissant le cosinus invariant est alors le suivant :

$$I_{d2}(Z) = \frac{u_i(z)}{u(z)} = \frac{\sin(\omega T_e)}{\omega(Z - \cos(\omega T_e))} \quad (1.51)$$

L'opérateur d'intégration ainsi obtenu étant strictement propre, il est retenu pour l'obtention de l'observateur discret. Notons que nous retrouvons par ce biais un intégrateur d'Euler de type **différences arrières** avec cependant la présence de coefficients de "**prewarping**" ou de mise à l'échelle qui assurent à la pulsation (ω) une totale adéquation entre l'intégrateur discret et son homologue continu puisque:

$$I_{d2}(Z) = I_{d2}(e^{(-j\omega T_e)}) = \frac{1}{j\omega} \quad (1.52)$$

Les coefficients de "**prewarping**" ont un autre avantage d'être paramétrés par la fréquence à privilégier, ce qui est un avantage si cette dernière fait partie du vecteur paramètre ou encore si elle est mesurable voire estimable. Pour la machine asynchrone, l'intégrateur est paramétré par la fréquence des tensions statoriques car c'est elle qui conditionne fortement la fréquence de ses signaux internes. Il en résulte par suite l'intégrateur suivant:

$$I_{d2}(q) = \hat{\omega}_{s,k}^{-1} \frac{\sin(\hat{\omega}_{s,k} T_e)}{\hat{\omega}_{s,k}(q - \cos(\hat{\omega}_{s,k} T_e))}, \hat{\omega}_{s,k} = \frac{1-q}{T_e} \tan^{-1} \frac{V_{s\beta,k}}{V_{s\alpha,k}} \quad (1.53)$$

Finalement, l'observateur à temps discret est obtenu par substitution de l'intégrateur continu $I_c(s) = s^{-1} I_n$ avec son homologue discret $I_d(q) = I_{d2}(q) \cdot I_n$.

Nous obtenons alors l'algorithme (1.54).

$$(Q_0) : \left\{ \begin{array}{l} \begin{bmatrix} \dot{x}_{0,k} \\ \hat{\phi}_{r\alpha,\beta,k} \end{bmatrix} = \begin{bmatrix} A_{0c} + \alpha_k A_{0v} & B_{0c} + \alpha_k B_{0v} \\ C_{0c} + \alpha_k C_{0v} & D_{0c} + \alpha_k D_{0v} \end{bmatrix} \begin{bmatrix} x_{0,k} \\ i_{s\alpha,\beta,k} \\ v_{s\alpha,\beta,k} \end{bmatrix} \\ x_{0,k} = \cos(\omega_{s,k} T_e) x_{0,k-1} + \frac{\sin(\omega_{s,k-1} T_e)}{\omega_{s,k-1}} \dot{x}_{0,k-1} \\ \omega_{s,k} = T_e \left(\tan^{-1} \frac{v_{s,\beta,k}}{v_{s\alpha,k}} \right) - \left(\tan^{-1} \frac{v_{s,\beta,k-1}}{v_{s\alpha,k-1}} \right) \\ \alpha_k = \frac{\underline{\Omega} - \underline{\Omega}_k}{\underline{\Omega} - \underline{\Omega}} - \end{array} \right. \quad (1.54)$$

Soit pour résumer pour l'observateur de flux, on a la description LFT suivante :

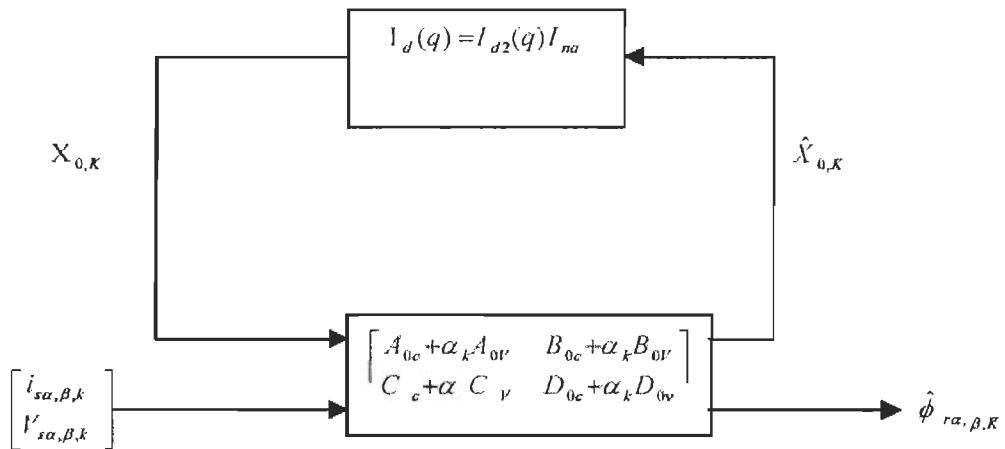


Figure 1.10 : Description LFT de l'observateur LPV discret

Discussion

Nous avons proposé une méthode permettant de discréteriser l'observateur synthétisé. Cette méthode a consisté à la substitution de l'opérateur d'intégration de la description LFT de l'observateur. L'intégrateur discret proposé est de type Euler avec des coefficients de "**prewarping**". Son avantage réside dans le fait qu'il n'introduit pas de bouclage algébrique ce qui est une alternative intéressante à la méthode proposée par Apkarian [21].

De plus, profitant du fait que la fréquence des signaux de l'observateur est conditionnée par la fréquence des tensions statoriques, les coefficients de "**prewarping**" sont adaptés en temps réel. Ceci permet d'assurer une parfaite intégration à la fréquence statorique et ce quelle que soit sa valeur. Les résultats et les formules présentés tout au long de ce paragraphe illustrent la justesse de la méthode proposée. Ils montrent en particulier qu'elle est parfaitement adaptée au contexte des systèmes LPV commandés par des signaux sinusoïdaux comme l'observateur de flux.

Conclusion

Ce chapitre nous a permis de faire une synthèse des travaux anciens et récents concernant les différentes méthodes d'observations avant de parler de l'estimateur du filtre de Kalman:

1. Problématiques liées à la commande et à l'observation (par principe de dualité).
2. L'observateur en boucle ouverte, l'observateur de Verghese ou de Kalman et les observateurs de type grands gains ou modes glissants.
3. Un exemple explicatif permet de comprendre l'observateur en boucle ouverte .
4. Synthèse d'un observateur LMI linéaire à paramètres variants.

Ce rappel assez bref nous a permis de bien nous situer dans l'environnement des estimateurs. Ce dernier était donné sous forme de définition utile à fin de faire une

comparaison entre les observateurs et d'apprécier de tel choix, de filtre de Kalman pour la suite de notre mémoire. Dans le prochain chapitre on parlera des outils pour l'analyse du filtre (EKF).

Chapitre 2

L'analyse du filtre de Kalman

Résumé : *Parmi l'ensemble des observateurs exposés au chapitre précédent, ce deuxième chapitre extrait l'un qui est principalement étudié au cours de ce mémoire. L'analyse de l'estimateur du filtre de Kalman de flux pour la machine asynchrone sera basée sur les définitions et les principes généraux qui conduisent à l'obtention d'un tel modèle ainsi que les problèmes liés à celui-ci.*

2.1. Introduction

Ce chapitre introduit les notions, les concepts, les définitions et les principes généraux du filtre de Kalman.

Dans un premier volet, on parlera des principes du filtrage de Kalman et des équations qui seront nécessaires pour notre étude ainsi que des problèmes de l'inobservabilité de l'état dans certaines conditions.

Dans un second volet, on abordera les problèmes d'implantation de l'estimation du flux d'une machine asynchrone et l'impact des erreurs numériques dues à la dissymétrie des matrices de variance-covariance sur le comportement du filtre. Par la suite on parlera des méthodes de réglage envisagées qui sont nécessaires à la bonne compréhension de ce mémoire.

L'approche retenue est basée sur la définition d'un critère qui exige moins de temps de calcul et, qui est robuste face aux variations paramétriques. Le problème de commande est alors reformulé comme un problème d'optimisation.

2.2. Principe du filtrage de Kalman

2.2.1. Généralités

De plus en plus utilisée dans les applications industrielles, la machine asynchrone est plus difficile à contrôler que la machine à courant continu et ce pour deux raisons. La première est que le modèle de la machine asynchrone est non linéaire, la seconde est que certaines grandeurs physiques sont difficilement mesurables (à moindre coût). Ces grandeurs de commande non accessibles sont le plus souvent estimées à partir d'un modèle d'état du système. L'adoption d'une approche déterministe, pour l'estimation d'état d'un système, suppose une connaissance exacte des paramètres du modèle du système considéré. De plus, ce type d'approche néglige les notions de fluctuations paramétriques et de fluctuations aléatoires de certaines grandeurs. Lorsque les bruits sont faibles, une approche déterministe peut s'avérer suffisante alors que l'obtention de performances optimisées peut être obtenue via l'utilisation des méthodes stochastiques.

Dans le cadre stochastique, le terme de filtre est préféré à celui d'observateur d'état plus communément utilisé dans le cadre de l'estimation déterministe bien que les structures soient totalement identiques.

L'avantage d'une approche stochastique est dans la corrélation précise qui existe entre le placement des pôles de l'estimateur et les paramètres statistiques des bruits.

En effet, pour un filtre de Kalman par exemple [1], le gain G de l'injection de sortie du filtre est calculé afin d'assurer le minimum de variance de l'erreur d'estimation.

Le but étant d'obtenir un filtre en vue d'une implémentation, il est plus intéressant de traiter directement la synthèse d'un filtre de Kalman discret. Les estimations issues de ce modèle sont sensibles aux erreurs de modélisation dues aux variations des paramètres du système.

En effet, aucun rebouclage avec les grandeurs réelles n'est possible permettant de prendre en compte les erreurs de modèle et les perturbations. De plus, la dynamique de l'estimateur est identique à celle du système réel. L'estimateur est donc dépendant, en régime dynamique, des modes naturels du système ce qui se caractérise par une convergence lente de l'erreur d'estimation. Par contre, ces estimateurs ne nécessitent qu'un faible coût de calcul algorithmique.

Un observateur est un estimateur possédant un rebouclage et dont la dynamique est différente de celle du système réel. Ainsi, il est possible de modifier la dynamique de l'observateur par un choix judicieux de la matrice de gain pour augmenter la vitesse de convergence de l'erreur d'observation vers zéro (en l'absence de perturbations). Les variations paramétriques et les perturbations peuvent être aussi prises en compte, puisque les sorties estimées sont comparées aux sorties mesurées et les écarts sont injectés à l'entrée de la matrice de gain pour réduire l'erreur d'observation.

2.2.2. Filtre de Kalman

2.2.2.1. Équations du filtre de Kalman

Notre but étant d'obtenir un filtre en vue d'une implémentation, il est plus intéressant de traiter directement la synthèse d'un filtre de Kalman discret. Soit (1.1) le modèle discret du système à observer est défini [9] par un modèle d'état à temps discret, composé de deux termes supplémentaires pour la prise en compte des bruits d'évolution ou d'état $V[K]$ et de mesure $W[K]$:

$$\begin{cases} X[K+1] = A_d X[K] + B_d U[K] + V[K] \\ Y[K+1] = C[K+1] X[K+1] + W[K+1] \end{cases} \quad (2.1)$$

Les bruits d'état et de mesure sont ici supposés centrés, blancs, indépendants et de matrices de variance-covariance Q et R , et ayant les propriétés statistiques suivantes:

Espérance mathématique nulle : $E(V[K])=0$

$$E(W[K])=0$$

Autocorrélation : $\forall \tau \in Z$

$$\begin{aligned} E(V[K] V[K-\tau]^t) &= Q \delta(\tau) \\ E(W[K] W[K-\tau]^t) &= R \delta(\tau) \end{aligned}$$

Absence de corrélation entre les bruits : $\forall \tau \in N^*$

$$E(W[K] V[K-\tau]^t) = 0$$

Absence de corrélation entre les bruits et l'état :

$$\begin{aligned} E(W[K] X[K-\tau]^t) &= 0 \\ E(V[K] X[K-\tau]^t) &= 0 \end{aligned}$$

L'estimateur $X[K+1]$ qui minimise la variance de l'erreur de prédition s'écrit [9] :

$$\begin{cases} X[K+1/K] = A_d X[K/K] + B_d U[K] \\ P[K+1/K] = A_d [K] P[K/K] A_d [K]' + Q \end{cases} \quad (2.2)$$

Cette prédition est ensuite corrigée par la nouvelle mesure de $Y[K+1]$:

$$\begin{cases} K[K+1] = P[K+1/K] C'[K+1] (C[K+1] P[K+1/K] C'[K+1] + R)^{-1} \\ X[K+1/K+1] = X[K+1/K] + K[K+1/K] (Y[K+1] - C[K+1] X[K+1/K]) \\ P[K+1/K+1] = P[K+1/K] - K[K+1] C[K+1] P[K+1/K] \end{cases} \quad (2.3)$$

Où le gain de Kalman $K[K+1]$ est calculé pour que l'erreur d'estimation soit statistiquement orthogonale à l'innovation $Y[K+1] - C[K+1] \hat{X}[K+1/K]$. Cette orthogonalité statistique permet de minimiser la variance de l'erreur d'estimation, lorsque les bruits sont gaussiens [9].

Le filtre de Kalman se base sur la connaissance du modèle d'état et du processus pour prédire le vecteur d'état d'une part, et les mesures réelles déjà faites pour corriger la prédition d'autre part. La prédition de l'état futur étant incertaine, les termes de la matrice de variance-covariance d'erreur d'estimation à priori $P[K+1/K]$ sont augmentés.

Cependant, la correction à partir des mesures réduit les termes de la matrice de variance-covariance d'erreur d'estimation $P[K+1/K+1]$. On notera que la matrice de gain est définie si la matrice $C_d P[K+1/K] C_d' + R$ est régulière. Ceci est toujours vérifié lorsque la matrice R est définie positive, c'est-à-dire, si toutes les sorties sont bruitées, ce qui est pratiquement toujours le cas en réalité. La matrice Q quant à elle doit être définie non négative. Dans le cas où l'on pose $Q=0$, le filtre de Kalman est un estimateur de la moyenne d'une variable aléatoire.

On peut faire l'analogie entre la commande et l'observabilité puisqu'elles sont des concepts duals. En effet, dans le cas de la commande, on cherche à annuler l'erreur entre la grandeur de référence et la grandeur commandée, tandis qu'un problème d'estimation consiste à annuler l'erreur entre les grandeurs mesurées et les grandeurs estimées.

L'algorithme du filtre de Kalman s'écrit [1] :

$$\begin{cases}
\tilde{X}[k+1|k] = A_d(\tilde{\Omega}) \tilde{X}[k|k] + B_d \tilde{U}[k] \\
\tilde{P}[k+1|k] = A_d(\tilde{\Omega}) \tilde{P}[k|k] A_d(\tilde{\Omega})^T + Q \\
\tilde{K}[k+1] = \tilde{P}[k+1|k] C \cdot (C \tilde{P}[k+1|k] C^T + R)^{-1} \\
\tilde{X}[k+1|k+1] = \tilde{X}[k+1|k] + \tilde{K}[k+1] \cdot (Y[k+1] - C \tilde{X}[k+1|k]) \\
\tilde{P}[k+1|k+1] = \tilde{P}[k+1|k] - \tilde{K}[k+1] C \tilde{P}[k+1|k] \\
\tilde{U}[k] = (U[k] + U[k+1])/2 \\
\tilde{Q}[k] = (Q[k] + Q[k+1])/2
\end{cases} \quad (2.4)$$

2.2.3. Estimateur étendu (non linéaire)

Pour l'estimation non-linéaire et affine en entrée U , les variables à estimer apparaissent à la fois dans le vecteur d'état et dans les matrices du modèle d'état. Le système d'équation d'état s'écrit [9]:

$$\begin{cases}
dX(t)/dt = f(X(t)) + g(X(t))U(t) \\
Y(t) = h(X(t))
\end{cases} \quad (2.5)$$

La résolution de ce type de problème se fait par la linéarisation des équations du système autour d'un point de fonctionnement, pour concevoir un algorithme d'estimation linéaire par la suite. Dans le but d'estimer les grandeurs internes où externes (perturbations, entrées inconnues,.....) du système, il est nécessaire d'augmenter l'ordre du modèle du nombre de grandeurs à estimer.

Ces dernières (notées par exemple Ω) peuvent avoir différentes variations au cours du temps (à l'échelle de la période d'échantillonnage). Elles peuvent varier très lentement variables vis-à-vis des autres grandeurs, on pose alors $\theta[K+1] = \theta$. Dans le cas où les variables évoluent au cours d'une période d'échantillonnage, on peut utiliser des équations d'évolution de plus en plus complexes.

Dans le cadre de l'observation du flux d'une machine asynchrone, le filtre de Kalman peut être utilisé suivant l'hypothèse que la dynamique de la vitesse est plus lente que celle des grandeurs électriques. Cette hypothèse est en effet importante car, cela permet de considérer la vitesse constante par morceaux vue des grandeurs électriques et en particulier celles du modèle électrique.

Pour l'estimation de la vitesse des machines électriques par exemple, le principe de séparation des grandeurs mécaniques et électriques n'est pas toujours valide, on pose l'équation continue: $d^2\theta(t)/dt = d\omega_m(t)/dt = P/J(C_{em} - C_r) - (F/J)\omega_m$ que l'on discrétise.

Les équations qui seront développées ci-dessous utilisent la première approximation afin d'employer par la suite les notions de découplage des états. On construit ainsi le modèle d'état augmenté discret suivant :

$$\begin{cases} \begin{bmatrix} X[K+1] \\ \theta[K+1] \end{bmatrix} = \begin{bmatrix} A_d(\theta[K]) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} X[K] \\ \theta[K] \end{bmatrix} + \begin{bmatrix} B_d(\theta[K]) \\ 0 \end{bmatrix} U[K] + V[K] \\ Y[K] = \begin{bmatrix} C(\theta[K]) & 0 \end{bmatrix} \begin{bmatrix} X[K] \\ \theta[K] \end{bmatrix} + W[K] \end{cases} \quad (2.6)$$

Lorsque les termes d'ordre supérieur ou égal à deux du développement limité de la fonction non linéaire $f(X)$ sont négligeables par rapport à la dérivée première, l'étape de prédiction s'écrit :

$$\begin{cases} \hat{X}[K+1/K] = A_d(\hat{\theta}[K/K])\hat{X}[K/K] + B_d(\hat{\theta}[K/K])U[K] \\ \hat{\theta}[K+1/K] = \hat{\theta}[K/K] \end{cases}$$

Un développement limité de la fonction [1] de transition autour de l'état estimé $X[K/K]$ conduit à une approximation de la matrice de variance-covariance de l'erreur de prédiction :

$$\begin{cases} P[K+1/K] = F[K]P[k/k]' + Q \\ \text{avec} \\ F[K] = \begin{bmatrix} A_d(\theta[K/K]) & \partial/\partial\theta(A_d[K]X[K/K] + B_d[K]U[K])_{\theta[K/K]} \\ 0 & I \end{bmatrix} \end{cases} \quad (2.7)$$

Ensuite, l'étape d'estimation correspond à une correction de l'état prédit à partir de l'innovation : $Y[K+1] - C[K+1]\hat{X}[K+1/K]$:

$$\hat{X}[K+1/K+1] = \hat{X}[K+1/K] + K[K+1](Y[K+1] - C[K+1]\hat{X}[K+1/K]) \quad (2.8)$$

Un développement limité au premier ordre de la fonction d'observation $h(X)$ autour de l'état prédit $X[K+1/K]$ fournit une approximation de la matrice de variance-covariance de l'erreur d'estimation :

$$\left\{ \begin{array}{l} P[K+1/K+1] = P[K+1/K] - K[K+1]H[K+1]P[K+1/K] \\ \text{avec} \\ H[K+1] = [Q[K+1] \quad \partial/\partial\theta(Q[K+1]X[K/K])_{\theta[K/K]}] \end{array} \right. \quad (2.9)$$

2.2.3.1. Observabilité du système augmenté

Parmi les points faibles des observateurs étendus, on trouve l'inobservabilité de l'état dans certaines conditions. Il est donc indispensable de faire un rappel avant de commencer la simulation.

2.2.3.2. Principe :

La méthode de l'analyse de l'observabilité des systèmes non linéaires consiste à dériver autant de fois qu'il est important la fonction de sortie. Ceci renseigne sur la possibilité d'exprimer les variables d'état en fonction des mesures (courants statoriques), des entrées (tensions statoriques) et de leurs dérivées.

Pour les systèmes non linéaires, l'observabilité est souvent réduite à un sous-espace de \mathbb{R}^n où n correspond à la dimension du vecteur d'état, qu'il est nécessaire de connaître afin d'éviter des modes de fonctionnement "interdits".

Les travaux [55] ont montré que l'estimation conjointe de la vitesse et de l'état électrique est possible en dehors des fonctionnements basses vitesses. Il conclut à la validité de ce résultat par simulation numérique quel que soit le repère choisi pour l'observation.

D'autres études effectuées en parallèle, démontrent mathématiquement que l'inobservabilité du système augmente lors d'un fonctionnement en génératrice et à basse vitesse dans le cas d'un modèle lié au repère statorique. Pour mieux comprendre les problèmes mis en jeu, nous allons détailler les résultats obtenus. Dans le cas des systèmes non linéaires, la notion d'observabilité complète n'existe pas et ça serait inutile d'en parler. Ici, l'observation peut être globale, locale et stationnaire.

Soit le système non linéaire continu défini par :

$$\begin{cases} \dot{X} = f(X, U) \\ Y = h(X, U) \end{cases} \quad (2.10)$$

Comme dans le cas des systèmes linéaires, l'observabilité est basée sur une dérivation des vecteurs de sorties Y.

$$\begin{cases} Y = h(X, U) \\ \dot{Y} = \frac{\partial h(X, U)}{\partial X} \frac{\partial X}{\partial t} + \frac{\partial h(X, U)}{\partial U} \frac{\partial U}{\partial t} \\ \vdots \\ Y^{(n-1)} = \frac{d^{n-1}h(X, U)}{dt^{n-1}} \end{cases}$$

où Y^n correspond à la dérivée n^{eme} de $h(X, U)$. Si nous posons pour simplifier les notations :

$Y^{[n-1]} = [Y \dots Y^{(n-1)}]$, $U^{[n-1]} = [U \dots U^{(n-1)}]$ et l'opérateur différentiel l'application N_f tel que :

$N_f^{[n-1]} = \left[\frac{\partial h(X, U)}{\partial X} \frac{\partial X}{\partial t} + \frac{\partial h(X, U)}{\partial U} \frac{\partial U}{\partial t} \right]$, l'application de l'observabilité peut s'écrire

de la façon suivante :

$$Y^{[n-1]} = \begin{bmatrix} N^0 \\ N \\ \vdots \\ N_f^{n-1} \end{bmatrix} h(X, U) = q(X, U^{[n-1]}) \quad (2.11)$$

2.2.3.1.2. Définition de l'observabilité globale:

Un système non linéaire est globalement observable si l'application d'observabilité (2.11) peut être résolue en fonction de X dans tout le domaine de définition, c'est-à-dire[10] :

$$\exists X = q^{-1}(Y^{[n-1]}, U^{[n-1]}) \quad \forall Y \in \mathbf{R}^m \quad \forall U \in \mathbf{R}^e$$

où m et e correspondent respectivement à la dimension des vecteurs Y et U.

L'inversion d'un système d'équations non linéaires est très délicate. Ainsi, les conditions d'inobservabilité sont difficilement obtenues de manière analytique. Par conséquent, l'analyse de l'observabilité est souvent réduite à un sous domaine en utilisant la notion d'observabilité locale.

2.2.3.1.3. Définition de l'observabilité locale:

L'analyse locale de l'observabilité d'un système non linéaire est définie grâce à un développement limité au premier ordre autour des coordonnées locales de [10] :

$$Y^{[n-1]} = q(X, U^{[n-1]}) + \left(\frac{\partial q}{\partial X} \right)_{X=X_0} (X - X_0) \text{ avec } J = \left(\frac{\partial q}{\partial X} \right)_{X=X_0}$$

Une condition nécessaire et suffisante pour l'observabilité locale est la régularité de la matrice jacobienne J, c'est-à-dire :

$$J = \left(\frac{\partial q}{\partial X} \right)_{X=X_0} = n \quad \forall U \in \mathbf{R}^m \Leftrightarrow \text{système non linéaire localement observable}$$

2.3. Problèmes d'implantation du filtre

2.3.1. Introduction

Lors d'une implantation d'un algorithme, les résultats obtenus sur un calculateur sont différents des résultats théoriques car, les calculateurs ne représentant qu'un ensemble fini de nombres (dépendant du nombre de bits, par exemple 8, 16 ou 32 bits), introduisent des erreurs d'arrondi et de troncature sur les résultats.

En effet, certains nombres dits rationnels ne sont pas représentables par un nombre fini de bits. Le nombre $1/3$ peut être décomposable en $1/4 + 1/16 + 1/64 + \dots$ s'écrit en base binaire $(0.010101010101\dots)_2$. Ainsi une partie de l'information est perdue pour enregistrer le résultat dans un registre de taille bien définie.

A titre indicatif, la majorité des DSP à virgule fixe ont des registres sur 16 bits, ainsi l'erreur d'arrondi introduite par le codage pour représenter le nombre $1/3$ est de $2.03.10^{-5}$. De plus, la plupart des DSP (Digital Signal Processor) à virgule fixe ont des accumulateurs sur 32 bits et leurs résultats sont stockés dans des cases mémoires de 16 bits. Des erreurs de troncature non négligeables peuvent ne pas avoir été prises en compte par le concepteur lors du choix et de l'écriture de l'algorithme.

Notre objectif principal lors de l'implantation est d'aboutir à des résultats présentant la ***Robustesse numérique*** qui est l'insensibilité du résultat à toutes sortes d'erreurs ou un minimum de ***Stabilité numérique*** qui est l'insensibilité du résultat aux erreurs d'arrondi uniquement. Comme nous venons de le voir, la précision d'un résultat dépend de la taille des cases mémoires du calculateur. Il est ainsi possible d'augmenter la dimension des emplacements mémoires afin de réduire les erreurs d'arrondi. Ceci s'effectue bien évidemment au détriment de la rapidité d'exécution de l'algorithme.

Un autre facteur important est la manière d'implémenter l'algorithme. Deux méthodes de calcul mathématiquement équivalentes peuvent produire des résultats bien différents pour une même précision de calcul.

2.3.2. Influence des erreurs numériques sur le comportement du filtre

Les erreurs numériques ne sont que les résultats de la dissymétrie des matrices de variance-covariance d'estimation à cause du mauvais conditionnement des matrices utilisées (matrice est une approximation) dans les équations du filtre de Kalman conventionnel. Ce qui provoque la dégradation des performances du filtre par une propagation des erreurs numériques dans les matrices de variance-covariance d'erreur d'estimation, et donc dans les gains de correction. Par la suite cette dégradation peut conduire à la divergence du filtre.

Certaines "symétrisations" des matrices de variance-covariance d'erreur d'estimation $P[K+1/K+1]$ améliorent le comportement du filtre [11]. En effet, le calcul de la matrice de variance-covariance d'erreur d'estimation n'aboutit pas forcément à une matrice symétrique, condition essentielle à la stabilité du filtre.

A titre d'exemple, le filtre de Kalman a été programmé sous deux implantations différentes pour l'estimation de la vitesse d'une machine asynchrone. La première forme ne comporte pas de "symétrisation". La deuxième force la nature de la matrice de variance-covariance d'erreur d'estimation $P[K+1/K+1]$ à être symétrique (figure 2.1):

$$(P[K+1/K+1] + P[K+1/K+1]')/2 \rightarrow P[K+1/K+1]$$

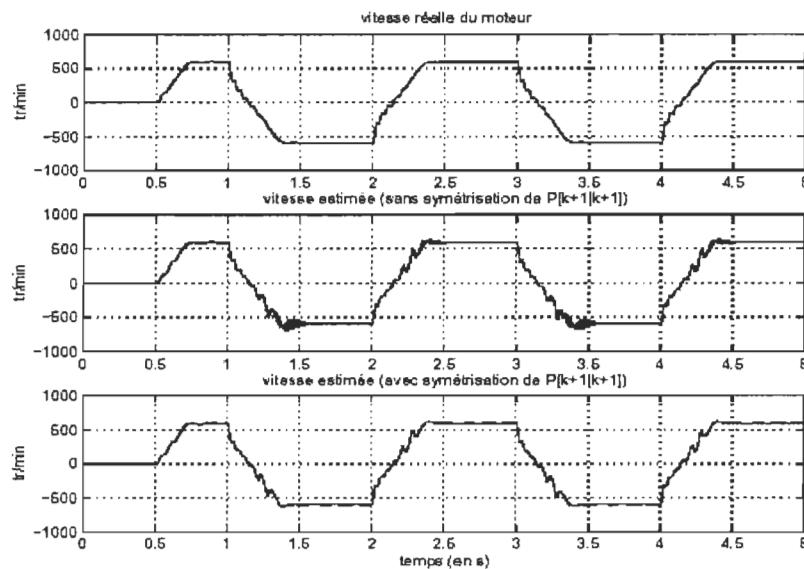


Figure 2.1: Apport d'une "symétrisation" des matrices de variance-covariance d'erreur d'estimation empruntée à [9].

2.3.3. Propagation des erreurs d'arrondi dans les filtres de Kalman

Le filtre de Kalman peut être décomposé en deux parties :

1. Une partie correspondant à la fois à la prédition et à l'estimation des états.
2. Une seconde partie correspondant aux calculs des matrices de variance-covariance d'erreur d'estimation et de prédition, ainsi que du gain.

L'étape d'estimation de l'état est un système bouclé. Les sorties de prédictions sont comparées aux mesures et les erreurs de prédition sont injectées dans la matrice afin de

corriger l'état. Si des erreurs de calcul numérique apparaissent sur les variables prédites, elles seront corrigées puisque ces prédictions sont comparées aux mesures du processus venant de l'extérieur. Ceci est évidemment valable si le gain de Kalman est correct.

La seconde partie ne comporte pas de bouclage et de lien avec l'extérieur. Il n'y a donc pas de référence externe permettant de corriger “l'estimation de $P[K+1/K+1]$ ”. Par conséquent, aucun moyen de détection et de correction des erreurs numériques introduites dans $P[K+1/K+1]$ n'est possible. De plus, le nombre d'opérations étant plus élevé que dans la première partie, la probabilité d'obtenir des erreurs numériques augmente.

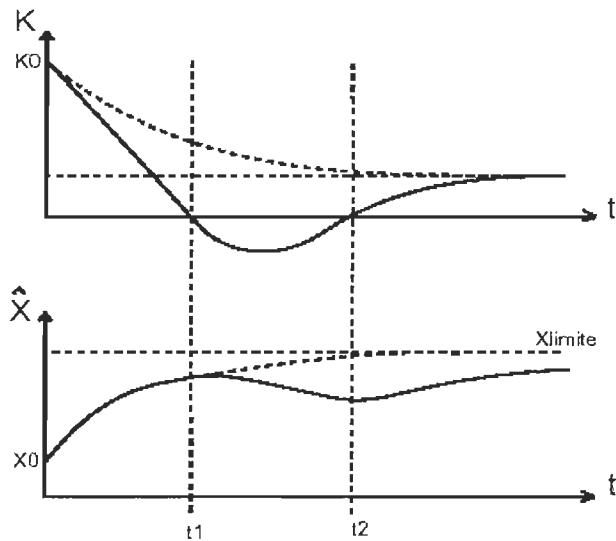


Figure 2.2 : Divergence temporaire du filtre de Kalman, figure empruntée à [12].

La propagation des erreurs d'arrondi peut conduire à la dégradation des performances du filtre ou à sa divergence. Une matrice $P[K+1/K]$ définie négative entraîne le changement du signe de K (cas monodimensionnel) car, elle est un facteur de la matrice de gain. Ainsi, au lieu de converger vers la bonne valeur comme schématisé sur la figure (2.2), le filtre diverge temporairement et peut retrouver une stabilité par la suite.

L'exemple illustratif montre que, momentanément, le gain calculé (trait plein) est de signe contraire à la valeur théorique (trait pointillé). Ce phénomène se traduit par une divergence du filtre. Lorsque le filtre redevient stable, la convergence est lente puisque le calcul du gain n'a pas pris en compte cette divergence temporaire. Le gain est donc plus faible qu'il ne devrait être.

2.4. Moyens de réglage du filtre de Kalman

Le filtre de Kalman se caractérise principalement par deux matrices Q et R qui contiennent la valeur de la covariance des bruits d'état et de mesure respectivement. Pour définir ces matrices, deux approches sont possibles :

1. La première approche consiste à analyser les erreurs de modélisation et de mesure. Elle est donc délicate à mettre en œuvre et suppose que ces erreurs puissent être représentées par des bruits blancs, centrés et decorrélates des états estimés, hypothèse rarement vérifiée. Une évaluation adéquate de ces erreurs permet un réglage quasi optimal du filtre et fournit aussi une évaluation des variances des états estimées. Elle nous renseigne sur la qualité de l'estimateur par une évaluation de la précision des estimations.
2. A l'inverse, la seconde approche consiste à choisir Q et R pour régler la dynamique de convergence des estimations. La plupart du temps, cela est fait de manière empirique (par essais successifs). Une approche équivalente repose sur l'automatisation du réglage de Q et R par la minimisation d'un critère déterministe [1].

Dans ce dernier cas, si les mesures sont de même nature avec des bruits de même importance, les termes de la diagonale de la matrice R sont souvent fixes à l'unité et le réglage de la dynamique du filtre s'effectue via les degrés de liberté α_1 à α_n [12].

$$R = I_m \quad Q = \begin{bmatrix} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \alpha_n \end{bmatrix}$$

Conclusion

Ce chapitre a permis de nous situer sur l'état de l'art et de faire une synthèse des travaux sur le filtrage de Kalman tels que :

1. Les problèmes liés à la résolution des équations matricielles du filtre et qui ont pour conséquence une modification du comportement du filtre .
2. L'inobservabilité de l'état dans certaines conditions par conséquent l'analyse de l'observabilité est souvent réduite à un sous-domaine (observabilité locale)
3. L'amélioration des performances algorithmiques pouvant se décomposer en trois catégories:
 1. Accroître la stabilité numérique du filtre par la réduction des erreurs d'arrondi.
 2. Réduction de la complexité algorithmique par optimisation du programme afin d'envisager son implantation sur des processeurs moins performants. Ou bien encore, conserver un processeur déjà employé et utiliser alors le temps gagné pour l'amélioration de la qualité des grandeurs reconstruites par l'estimation des paramètres et/ou des perturbations (entrées inconnues).
 3. Les moyens de réglage efficace du filtre de Kalman soit par analyse des erreurs de modélisation et de mesure, soit choisir Q et R pour régler la dynamique de convergence des estimations.

Chapitre 3

Méthodologie pour la reconstruction du flux rotorique d'une machine asynchrone

Résumé : Ce chapitre propose un observateur de flux de type EKF pour la machine asynchrone. Les outils utilisés pour le synthétiser sont ceux précédemment présentés au chapitre 2. Après la modélisation, l'observateur résultant est testé en simulation selon trois discréétisations différentes.

Les résultats ainsi obtenus sont comparés à ceux de l'observateur de référence réel choisi comme observateur étalon.

3.1. Introduction

Le chapitre précédent a présenté les outils pour la synthèse du filtre de Kalman. Or, comme pour la synthèse du filtre de Kalman, la mise en œuvre de ces résultats passe par la définition du modèle standard permettant de spécifier les différents objectifs de commande qui seront traité dans ce chapitre.

Certaines grandeurs physiques sont difficilement mesurables (à moindre coût). Les méthodes de reconstitution indirecte du flux, utilisant uniquement les grandeurs accessibles (courants statoriques) et le modèle de la machine asynchrone, sont préférées aux méthodes de mesure directe (capteurs à effet Hall par exemple) pour leur fiabilité, leur faible coût d'entretien et leur faible sensibilité aux bruits de mesures. En effet les constructeurs de variateur de vitesse tendent à diminuer le nombre de capteurs et les supprimer dans quelques applications si possible.

Parmi les méthodes de la reconstruction du flux rotorique, on trouve le filtrage de Kalman. Il nous permet d'aboutir à deux points délicats d'une façon optimale tel que la réduction du coût de calcul du filtre et le réglage des matrices afin d'effectuer l'observation du flux de la machine asynchrone.

3.2. Modélisation du système considéré

3.2.1. Généralités

Dans la très grande majorité des applications nécessitant l'utilisation d'une machine asynchrone à vitesse variable, cette dernière voit son flux contrôlé. En effet, le contrôle du flux permet à la machine de fournir le maximum de couple ce qui lui confère de très bonnes performances tant statiques que dynamiques. Cependant, le flux n'est pas mesurable à moindre coût; c'est pourquoi il faut l'estimer. La méthode de Park, couramment employée pour modéliser une machine asynchrone, ne conduit pas à un seul et unique modèle. Le modèle le plus utilisé est exprimé dans un repère diphasé quelconque. Les variables d'état constituant ce modèle sont les courants et tensions statoriques ($i_{sd,qx}$, $v_{sd,qx}$), le flux rotorique ($\Phi_{rd,qx}$), la position et ou la vitesse du rotor (Ω, θ).

$$\begin{bmatrix} \dot{i}_{sd,qx} \\ \dot{\Phi}_{rd,qx} \\ \dot{\theta} \\ \dot{\Omega} \end{bmatrix} = f(i_{sd,qx}, \Phi_{rd,qx}, \Omega, \theta, V_{d,qx}, T) \quad (3.1).$$

3.2.2. Mise en équations électriques et magnétiques

Tout d'abord nous rappelons que l'ensemble rotor/stator est constitué de deux systèmes triphasés équilibrés, et que la répartition de l'induction dans l'entrefer est supposée sinusoïdale. Les six enroulements sont décrits par les équations suivantes [14-16] :

$$\text{Équation statoriques} \quad \begin{cases} V_{sa} = R_s i_{sa} + \frac{d\Phi_{sa}}{dt} \\ V_{sb} = R_s i_{sb} + \frac{d\Phi_{sb}}{dt} \\ V_{sc} = R_s i_{sc} + \frac{d\Phi_{sc}}{dt} \end{cases} \quad (3.2)$$

$$\text{Équation rotoriques} \quad \begin{cases} V_{ra} = R_r i_{ra} + \frac{d\Phi_{ra}}{dt} \\ V_{rb} = R_r i_{rb} + \frac{d\Phi_{rb}}{dt} \\ V_{rc} = R_r i_{rc} + \frac{d\Phi_{rc}}{dt} \end{cases} \quad (3.3)$$

1. V_{si} et V_{rj} : correspondent aux tensions statoriques et rotoriques $i, j = \{a, b, c\}$.
2. i_{si} et i_{rj} : correspondent aux courants dans les enroulements statoriques et rotoriques.
3. Φ_{si} et ϕ_{rj} : correspondent aux flux à travers les enroulements statoriques et rotoriques.
4. R_s et R_r : correspondent aux résistances statoriques et rotoriques.

On remarque que l'entrefer est constant. Le stator et le rotor étant triphasés et de construction symétrique. Les inductances propres et mutuelles entre phases d'un même enroulement sont constantes et égales. On désigne par :

1. L_{ps} : inductances propres statoriques.
2. L_{pr} : inductances propres rotoriques.
3. M_s : inductances mutuelles entre deux phases statoriques.
4. M_r : inductances mutuelles entre deux phases rotoriques.
5. M : inductance mutuelle maximale entre une phase statorique et rotorique.
6. θ : angle électrique instantané entre les axes magnétiques des phases du stator et du rotor.

D'après la figure (3.1), les inductances mutuelles entre phases du stator et du rotor sont fonction de l'angle θ et s'expriment de manière générale comme suit:

$$M_{si-rj} = M \cos(\theta + (j-i) \frac{2\pi}{3})$$

Où i est l'indice des phases du stator et j l'indice des phases du rotor. D'où les expressions des flux sous forme d'équation matricielle :

$$\begin{bmatrix} \phi_s \\ \phi_r \end{bmatrix} = [L(\theta)] \begin{bmatrix} i_s \\ i_r \end{bmatrix} \quad (3.4)$$

Les flux magnétiques sont donnés par l'équation matricielle :

$$\begin{bmatrix} \phi_{sa} \\ \phi_{sb} \\ \phi_{sc} \\ \phi_{ra} \\ \phi_{rb} \\ \phi_{rc} \end{bmatrix} = \begin{bmatrix} L_{ps} & M_s & M_s & M_{sa-ra} & M_{sa-rb} & M_{sa-rc} \\ M_s & L_{ps} & M_s & M_{sb-ra} & M_{sb-rb} & M_{sa-rc} \\ M_s & M_s & L_{ps} & M_{sc-ra} & M_{sc-rb} & M_{sc-rc} \\ M_{ra-sa} & M_{ra-sb} & M_{ra-sc} & L_{pr} & M_r & M_r \\ M_{rb-sa} & M_{rb-sb} & M_{rb-sc} & M_r & L_{pr} & M_r \\ M_{rc-sa} & M_{rc-sb} & M_{rc-sc} & M_r & M_r & L_{pr} \end{bmatrix} \begin{bmatrix} i_{sa} \\ i_{sb} \\ i_{sc} \\ i_{ra} \\ i_{rb} \\ i_{rc} \end{bmatrix} \quad (3.5)$$

avec :

1. $M_{sa-ra} = M_{sb-rb} = M_{sc-rc} = M \cos(\theta)$
2. $M_{sa-rb} = M_{sa-rc} = M_{sc-ra} = M \cos(\theta + \frac{2\pi}{3})$
3. $M_{sa-rc} = M_{sc-rb} = M_{sb-ra} = M \cos(\theta - \frac{2\pi}{3})$
4. $M_{ra-sa} = M_{rb-sb} = M_{rc-sc} = M \cos(\theta)$
5. $M_{ra-sb} = M_{ra-sc} = M_{rc-sa} = M \cos(\theta - \frac{2\pi}{3})$
6. $M_{ra-sc} = M_{rc-sb} = M_{rb-sa} = M \cos(\theta + \frac{2\pi}{3})$

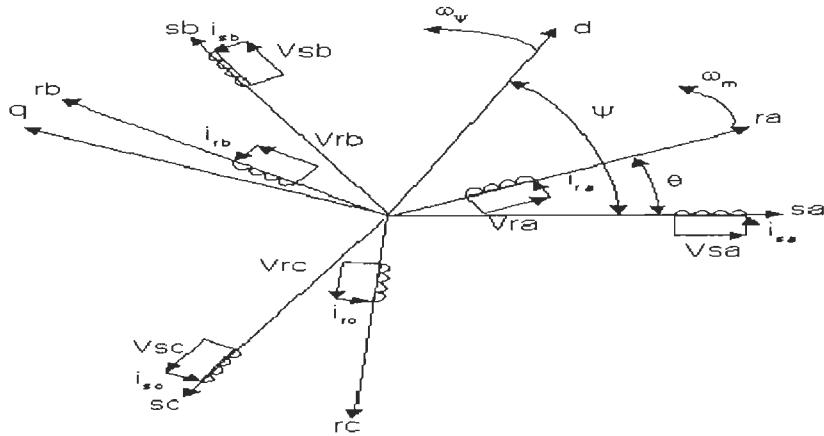


Figure 3.1: Modèle de la machine asynchrone dans le repère de Park.

Les tensions triphasées du stator V_s , les courants statoriques i_s et les courants rotoriques i_r s'écrivent en tenant compte des expressions des flux et en désignant par p l'opérateur de dérivation d/dt .

Les équations des tensions au stator et au rotor s'écrivent sous forme matricielle :

$$\begin{cases} [V_s] = [R_s][I_s] + p[L_{ss}][i_s] + [M_{si-rj}][i_r] \\ [V_r] = [R_r][I_r] + p[L_{rr}][i_r] + [M_{rj-si}][i_s] \end{cases} \quad (3.6)$$

avec :

$$\begin{aligned} V_s &= [V_{sa}, V_{sb}, V_{sc}]^t & V_r &= [V_{ra}, V_{rb}, V_{rc}]^t \\ i_s &= [i_{sa}, i_{sb}, i_{sc}]^t & i_r &= [i_{ra}, i_{rb}, i_{rc}]^t \\ [R_s] &= R_s I_3 & [R_r] &= R_r I_3 & \text{et } I_3 &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

$$[L_{ss}] = \begin{bmatrix} L_{ps} & M_s & M_s \\ M_s & L_{ps} & M_s \\ M_s & M_s & L_{ps} \end{bmatrix} \quad [L_{pr}] = \begin{bmatrix} L_{pr} & M_r & M_r \\ M_r & L_{pr} & M_r \\ M_r & M_r & L_{pr} \end{bmatrix}$$

$$[M_{si-rj}] = [M_{rj-si}]^t = M \begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{4\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) \\ \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) & \cos(\theta - \frac{4\pi}{3}) \\ \cos(\theta - \frac{4\pi}{3}) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta) \end{bmatrix}$$

3.2.3. Transformation de Park

Afin de simplifier le modèle précédent, on peut réaliser un changement de base sur les grandeurs physiques (tension, flux et courant) qui conduit à une relation matricielle indépendante de l'angle θ . Ce changement de base est appelé transformation de Park qui transforme les enroulements statoriques a, b et c en des enroulements équivalents du point de vue électrique et magnétique et disposés selon deux axes d et q perpendiculaires. Le référentiel ainsi obtenu est appelé (d,q) pour directe et quadratique.

L'avantage d'utiliser un référentiel est d'aboutir à un modèle qui manipule des grandeurs continues.

Les inductances mutuelles entre ces enroulements sont variables au cours du temps et fonction de l'angle électrique. La matrice reliant les flux aux courants fait intervenir des coefficients en fonction de l'angle électrique θ , ceci complique notablement l'étude des régimes transitoires.

3.2.3.1. Modèle dans le plan diphasé($\alpha-\beta$) équivalent

Les équations de la machine dans un repère diphasé fixe par rapport au stator, sont obtenues par application de la transformation linéaire de Concordia T_{3-3} correspondant à l'invariance de la puissance dans la transformation aux équations électriques vue auparavant. Le modèle équivalent est la réduction d'ordre des équations électriques décrivant le système machine asynchrone. On passe d'un ordre six à un ordre quatre. Le système de coordonnées a-b-c est ainsi transformé en un système de coordonnées $\alpha-\beta$.

$$\begin{bmatrix} x_0 \\ x_\alpha \\ x_\beta \end{bmatrix} = T_{3-3} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad T_{3-3} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \quad (3.7)$$

Comme l'alimentation triphasée étant équilibrée, la somme des courants $\sum i_{a,b,c}$ est nulle. Nous vérifions d'après l'équation (3.7) que $i_0 = 0$. Le courant homopolaire et le flux homopolaire s'annulent. Par conséquent, cet axe est rarement pris en compte dans la transformation triphasée diphasée. Il se résume à :

$$\begin{bmatrix} x_\alpha \\ x_\beta \\ x_\gamma \end{bmatrix} = T_{3 \rightarrow 2} \begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix} \quad T_{3 \rightarrow 2} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \quad (3.8)$$

La transformation inverse est directement tirée de (3.7). On a $x_{abc} = T_{3 \rightarrow 3}^{-1} x_{\alpha\beta\gamma}$.

La matrice de transformation est orthogonale donc $T_{3 \rightarrow 3}^{-1}$.

Pour exprimer toutes les grandeurs dans un même repère, les grandeurs statoriques et rotoriques sont projetées dans un repère tournant d-q décalé de ψ par rapport au repère fixe noté $\alpha - \beta$. Cette transformation se fait à partir de la matrice orthogonale de rotation T_ε .

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = T_\varepsilon \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} \quad T_\varepsilon = \begin{bmatrix} \cos(\varepsilon) & \sin(\varepsilon) \\ -\sin(\varepsilon) & \cos(\varepsilon) \end{bmatrix} \quad (3.9)$$

L'angle de rotation ε est donné par $\varepsilon = \psi$ pour les grandeurs statoriques et $\varepsilon = \psi - \theta$ pour les grandeurs rotoriques. Le repère d-q n'est lié, pour le moment, à aucun référentiel particulier. Nous verrons qu'il existe trois repères privilégiés pour la commande et l'observation. La matrice de rotation est une matrice orthogonale donc $T_\varepsilon^{-1} = T^t \varepsilon$.

Les deux transformations peuvent être réunies en une seule transformation T définie comme suit :

$$T = T_\varepsilon T_{3 \rightarrow 2} = \sqrt{\frac{2}{3}} \begin{pmatrix} \cos(\varepsilon) & \cos(\varepsilon - 2\pi/3) & \cos(\varepsilon - 4\pi/3) \\ -\sin(\varepsilon) & -\sin(\varepsilon - 2\pi/3) & -\sin(\varepsilon - 4\pi/3) \end{pmatrix} \quad (3.10)$$

On a ainsi : $[i_{dq}] = [T][i_{abc}]$ $[v_{dq}] = [T][v_{abc}]$ $[\Phi_{dq}] = [T][\Phi_{abc}]$

Le rôle du facteur $\sqrt{2/3}$ est de ne pas modifier la puissance consommée par la machine lors de la transformation. Lorsque ce changement de base est réalisé sur les variables tension et courant, la valeur de la puissance reste constante (le produit scalaire tension \times courant est indépendant de la base). Par la suite, c'est cette transformation T qui sera utilisée afin de conserver l'équivalence énergétique.

Remarque: lorsque la valeur zéro est attribuée à l'angle ϵ , la transformation de Park particularisée porte le nom de transformation de Concordia (équation (3.3)) et les axes d-q sont alors désignés par les axes $\alpha - \beta$.

3.2.4. Équations de Park de la machine asynchrone dans les axes dq

L'application du changement de base T (expression 3.10) sur les équations électriques et magnétiques (3.1) conduit à l'équation suivante :

$$[V_{dq}] = [T][V_{abc}] = [T][R_s][i_{abc}] + [T]\frac{d\Phi_{abc}}{dt}$$

Puis, en utilisant les transformées inverses :

$$[V_{dq}] = [T][R_s][T]^{-1}[i_{dq}] = [T][T]^{-1}\frac{d[\Phi_{dq}]}{dt} + [T]\frac{d[T]^{-1}}{dt}\Phi_{dq} \quad (3.11)$$

On remarque que : $[T][T]^{-1} = 1$ et $[T]\frac{d[T]^{-1}}{dt} = \frac{d\epsilon}{dt} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

Par application des propriétés des matrices de Park en développant l'équation (3.11) et en tenant compte des deux relations précédentes, on obtient les équations de Park (le rotor étant en court-circuit, les tensions V_{rd} et V_{rq} sont nulles) :

Équations statoriques

$$\begin{cases} V_{sd} = R_s I_{sd} + \frac{d\Phi_{sd}}{dt} - \omega_\psi \Phi_{sq} \\ V_{sq} = R_s I_{sq} + \frac{d\Phi_{sq}}{dt} + \omega_\psi \Phi_{sd} \end{cases} \quad (3.12)$$

Équations rotoriques

$$\begin{cases} 0 = R_r I_{rd} + \frac{d\Phi_{rd}}{dt} - (\omega_\psi - \omega_m) \Phi_{rq} \\ 0 = R_s I_{rq} + \frac{d\Phi_{rq}}{dt} + (\omega_\psi - \omega_m) \Phi_{rd} \end{cases} \quad (3.8)$$

avec:

1. ω_ψ : vitesse de rotation du repère tournant d-q : $\omega_\psi = \frac{d\psi}{dt}$.
2. ω_m : pulsation mécanique ramenée à la fréquence des grandeurs électriques ($\omega_m = P\Omega$), $\omega_m = \frac{d\theta}{dt}$
3. P : nombre de paires de pôles.

L'application de la transformation de Park sur les grandeurs magnétiques simplifie les équations matricielles et ces équations ne sont plus fonction de l'angle θ .

$$\begin{bmatrix} \Phi_{sd} \\ \Phi_{rd} \end{bmatrix} = \begin{bmatrix} L_s & M_{sr} \\ M_{sr} & L_r \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{rd} \end{bmatrix} \quad \begin{bmatrix} \Phi_{sq} \\ \Phi_{rq} \end{bmatrix} = \begin{bmatrix} L_s & M_{sr} \\ M_{sr} & L_r \end{bmatrix} \begin{bmatrix} i_{sq} \\ i_{rq} \end{bmatrix} \quad (3.9)$$

Où:

1. L_s : inductance cyclique statorique ($L_s = L_{ps} - M_s$)
2. L_r : inductance cyclique rotorique ($L_r = L_{pr} - M_r$)
3. M_{sr} : inductance mutuelle cyclique entre stator et rotor ($M_{sr} = \frac{3}{2} M$)

Enfin une expression générale du couple électromagnétique dans ce même repère est :

$$C_{em} = P \frac{M_{sr}}{L_r} (\Phi_{rd} I_{sq} - \Phi_{rq} I_{sd}) \quad (3.10)$$

Afin de concevoir le modèle complet, il faut ajouter l'équation mécanique :

$$C_{em} - C_r = J \frac{d\Omega}{dt} + f\Omega \quad (3.11)$$

d'où, en utilisant la vitesse de rotation électrique ω_m ($\omega_m = P\Omega$), on obtient :

$$\frac{d\omega_m}{dt} = -\frac{f}{J} \omega_m + \frac{P}{J} (C_{em} - C_r) \quad (3.12)$$

Les équations (3.7), (3.8), (3.9), (3.10) et (3.12) constituent le modèle de base de la machine asynchrone.

3.2.5. Système (d'état) non linéaire

En vue de la commande, la forme des équations précédentes (sous forme d'équations d'état) permettent d'obtenir des trajectoires d'état qui, en régime permanent sont constantes, et de se rapprocher au plus d'un comportement statique et dynamique équivalent à celui d'une MCC. Cette forme demande dans un premier temps la définition du vecteur d'état X , du vecteur de commande U et du vecteur de sortie Y . Le flux étant difficilement mesurable sur une machine asynchrone, le vecteur de sortie sera composé des courants statoriques $i_{s\alpha}$ et $i_{s\beta}$ ainsi que de la vitesse de rotation du rotor Ω dans le cas d'une commande avec capteur de vitesse. Le vecteur d'état X est constitué des deux courants statoriques i_{sd} et i_{sq} qui conduisent à une matrice de mesure constante dans le cas d'une modélisation dans le repère statorique. Puisque nous tenons compte du couplage des modes électrique et mécanique, le vecteur X contient la vitesse de rotation ω_m . Il reste deux composantes à choisir. Nous prenons les deux composantes du flux rotorique.

Le vecteur d'état X , le vecteur d'entrée U et le vecteur de sortie Y sont alors définis de la manière suivante :

$$X_{dq} = \begin{bmatrix} i_{sd} \\ i_{sq} \\ \Phi_{rd} \\ \Phi_{rq} \\ \omega_m \end{bmatrix} \quad U = \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad Y = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Omega \end{bmatrix} \quad (3.13)$$

En remplaçant les courants rotoriques et les flux statoriques dans (3.7) et (3.8) par les expressions (3.9) et le couple électromagnétique dans (3.12) par (3.10), on obtient l'équation d'état suivante :

$$\left\{ \begin{array}{l} \frac{dX_{dq}}{dt} = f(X_{dq}, \omega_\Psi, C_r) + g(\Psi)U \\ \text{et} \\ Y = h(\Psi)X_{dq} \end{array} \right. \quad (3.14)$$

Où :

$$\frac{d}{dt} X_{dq} = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)i_{sd} + \omega_\Psi i_{sq} + \frac{1-\sigma}{\sigma} \frac{1}{M_{sr} T_r} \Phi_{rd} + \frac{1-\sigma}{\sigma M_{sr}} \omega_m \Phi_{rq} \\ -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)i_{sq} - \omega_\Psi i_{sd} + \frac{1-\sigma}{\sigma} \frac{1}{M_{sr} T_r} \Phi_{rq} - \frac{1-\sigma}{\sigma M_{sr}} \omega_m \Phi_{rd} \\ \frac{M_{sr}}{T_r} i_{sd} - \frac{1}{T_r} \Phi_{rd} + (\omega_\Psi - \omega_m) \Phi_{rq} \\ \frac{M_{sr}}{T_r} i_{sq} - \frac{1}{T_r} \Phi_{rq} - (\omega_\Psi - \omega_m) \Phi_{rd} \\ \frac{P^2 M_{sr}}{J L_r} \Phi_{rd} i_{sq} - \frac{P^2 M_{sr}}{J L_r} \Phi_{rq} i_{sd} - \frac{f}{J} \omega_m - \frac{P}{J} C_r \end{bmatrix} + \begin{bmatrix} \frac{\cos(\Psi)}{\sigma L_s} & \frac{\sin(\Psi)}{\sigma L_s} \\ \frac{\sin(\Psi)}{\sigma L_s} & \frac{\cos(\Psi)}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad (3.15)$$

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Omega \end{bmatrix} = \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 & 0 & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{P} \end{bmatrix} \begin{bmatrix} i_{sd} \\ i_{sq} \\ \Phi_{rd} \\ \Phi_{rq} \\ \omega_m \end{bmatrix} \quad (3.16)$$

avec :

- $\sigma = 1 - \frac{M_{sr}^2}{L_s L_r}$: le coefficient de dispersion ou de Blondel
- $T_r = \frac{L_r}{R_r}$: la constante de temps rotorique

Le modèle de la MAS obtenue par la transformation de Park est non linéaire et de degré de complexité comparable à un système différentiel non linéaire affine en entrée représenté par l'équation:

$$\left\{ \begin{array}{l} \frac{dX_{dq}}{dt} = f(X_{dq}, C_r) + g(X_{dq})U \\ Y = h(X_{dq}) \end{array} \right. \quad (3.17)$$

En introduisant les constantes simplificatrices suivantes :

$$a = \frac{1}{\sigma L_s} \quad b = \frac{1}{\sigma L_s} \quad c = \frac{1-\sigma}{\sigma M_{sr}} \quad k_m = \frac{P^2 M_{sr}}{J L_r} \quad m_1 = \frac{f}{J} \quad m_2 = \frac{P}{J} \quad (3.18)$$

on obtient :

$$\frac{d}{dt} X_{dq} = \begin{bmatrix} -(aR_s + c \frac{M_{sr}}{L_r} R_r) i_{s\alpha} + \frac{c}{L_r} R_r \Phi_{r\alpha} + c \omega_m \Phi_{r\beta} \\ -(aR_s + c \frac{M_{sr}}{L_r} R_r) i_{s\beta} + \frac{c}{L_r} R_r \Phi_{r\beta} - c \omega_m \Phi_{r\alpha} \\ \frac{M_{sr}}{L_r} R_r i_{s\alpha} - \frac{1}{L_r} R_r \Phi_{r\alpha} - \omega_m \Phi_{r\beta} \\ \frac{M_{sr}}{L_r} R_r i_{s\beta} - \frac{1}{L_r} R_r \Phi_{r\beta} + \omega_m \Phi_{r\alpha} \\ K_m \Phi_{r\alpha} i_{s\beta} - K_m \Phi_{r\beta} i_{s\alpha} - m_1 \omega_m - m_2 C_r \end{bmatrix} + \begin{bmatrix} a & 0 \\ 0 & a \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad (3.19)$$

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Omega \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 & 0 & 0 \\ \sin(\psi) & \cos(\psi) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{P} \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Phi_{r\alpha} \\ \Phi_{r\beta} \\ \omega_m \end{bmatrix} \quad (3.20)$$

Le modèle non linéaire affine dans le repère statorique $\alpha-\beta$ résulte directement du modèle $d-q$ en posant $\omega_\psi = 0$ et $\psi = 0$ d'où :

$$\begin{cases} \frac{dX_{\alpha\beta}}{dt} = f(X_{\alpha\beta}, C_r) + BU \\ Y = CX_{\alpha\beta} \end{cases} \quad (3.21)$$

$$\text{soit : } \frac{d}{dt} X_{\alpha\beta} = \begin{bmatrix} -(aR_s + c \frac{M_{sr}}{L_r} R_r) i_{s\alpha} + \frac{c}{L_r} R_r \Phi_{r\alpha} + c \omega_m \Phi_{r\beta} \\ -(aR_s + c \frac{M_{sr}}{L_r} R_r) i_{s\beta} + \frac{c}{L_r} R_r \Phi_{r\beta} - c \omega_m \Phi_{r\alpha} \\ \frac{M_{sr}}{L_r} R_r i_{s\alpha} - \frac{1}{L_r} R_r \Phi_{r\alpha} - \omega_m \Phi_{r\beta} \\ \frac{M_{sr}}{L_r} R_r i_{s\beta} - \frac{1}{L_r} R_r \Phi_{r\beta} + \omega_m \Phi_{r\alpha} \\ K_m \Phi_{r\alpha} i_{s\beta} - K_m \Phi_{r\beta} i_{s\alpha} - m_1 \omega_m - m_2 C_r \end{bmatrix} + \begin{bmatrix} a & 0 \\ 0 & a \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad (3.22)$$

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Omega \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{P} \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Phi_{r\alpha} \\ \Phi_{r\beta} \\ \omega_m \end{bmatrix} \quad (3.23)$$

Remarque :

Le modèle de la machine asynchrone obtenu est non linéaire. Pour commander de tels systèmes, il est recommandé si possible de développer des commandes linéaires à partir d'une approximation linéaire de leur modèle au détriment des performances obtenues.

3.2.6. Choix du référentiel

Il est important de noter que le choix du référentiel et les transformations triphasé/biphasé permettent d'obtenir une première simplification de l'écriture des équations d'état. Ceci implique que le choix du type de commande à appliquer dépendra bien du plan où on décide de travailler.

Trois référentiels sont possibles en vue de l'observation. Ils dépendent de l'orientation de l'axe d du repère de Park :

1. Modèle dans le repère fixe $\alpha - \beta$ lié au stator, $\psi = 0$.
2. Modèle dans le repère tournant d-q lié au flux rotorique, $\psi = \arg(\phi_{rd} + j\phi_{rq})$
3. Modèle dans le repère tournant d-q lié au rotor, $\psi = \theta$.

3.2.6.1. Lieux des pôles pour les trois repères

Un premier choix de référentiel en vue de l'observation [53] est obtenu à partir de l'étude des lieux des pôles pour les trois repères en fonction du paramètre externe ω_m .

Les valeurs propres de la matrice de transition $f(X_{dq})$ permettent de déterminer le comportement en régime transitoire par évaluation des pôles du système pour les trois référentiels possibles.

La pulsation des grandeurs statoriques est égal à 100π rad/s et les points marqués par "a" sur les courbes correspondent à $\omega_m=0$ rad/s et ceux marqués par "b" correspondent à $\omega_m=100\pi$ rad/s.

L'étude des trois courbes des figures (3.2), (3.3) et (3.4) montre que, quand ω_m tend vers zéro, les valeurs propres dominantes ont des parties réelles faibles, caractéristiques des systèmes lents.

L'étude des systèmes liés au stator ou au rotor ont aussi montré qu'ils ont des parties imaginaires de faible amplitude à faible vitesse par rapport au système lié au champ tournant. Ils se caractérisent par un comportement dominant de type peu oscillatoire.

Le choix du repère rotorique par rapport au repère fixe est un modèle mieux amorti sur toute la plage de vitesse puisque les parties imaginaires des valeurs propres dominantes restent inférieures à 75 rad/s.

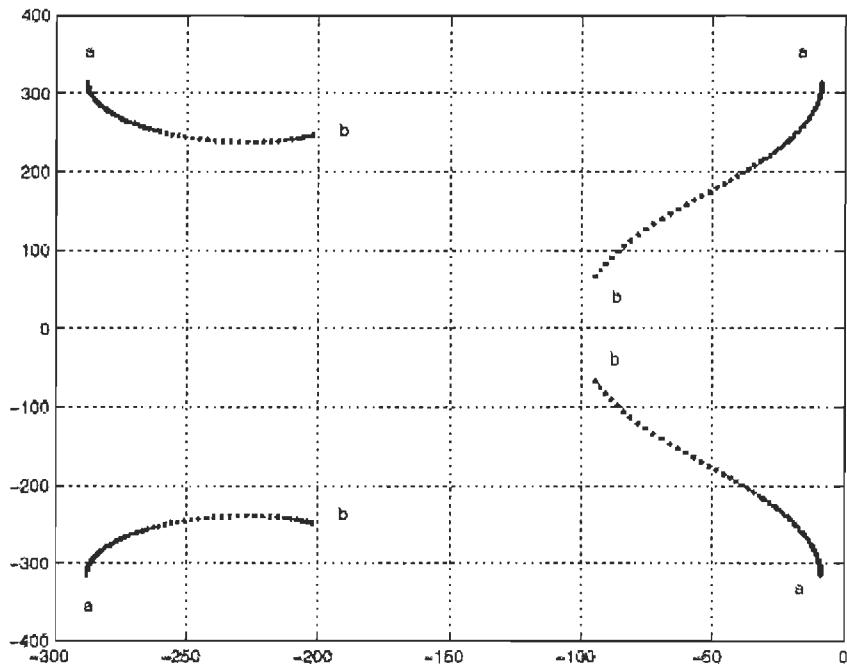


Figure 3.2: lieu des pôles dans un repère lié au champ tournant.

Par contre, le choix d'un autre repère exige l'utilisation de matrices de transformation trigonométriques dépendant de la position instantanée du rotor, ce qui augmente donc la complexité de l'algorithme afin de "ramener" les grandeurs électriques au rotor. Ces premières constatations nous permettent de dire que le choix d'un repère lié au champ tournant a les mêmes inconvénients algorithmiques que le repère lié au rotor et conduit de plus à un système plus oscillatoire dès les basses vitesses.

Dans le repère fixe $\alpha - \beta$, l'axe d est confondu avec l'axe α du stator et les grandeurs varient de manière alternative à la fréquence ω_s . Les conditions qui déterminent ce

référentiel sont : $\omega_\psi = \omega_s = 0$ et $\frac{d\theta_{sl}}{dt} = \omega_{sl} = -\omega_m$.

L'observation des grandeurs alternatives dans les axes $\alpha - \beta$ exige une intégration précise qui peut être obtenue avec une faible période d'échantillonnage T_e et un DSP pour l'implantation numérique. L'angle ψ de la transformation T peut être estimé directement à partir des composantes du vecteur d'état observé. Par exemple, l'angle de transformation pour l'orientation du repère au champ tournant en vue de la commande se calcule à partir des composantes du flux rotorique observées $\hat{\phi}_{r\alpha}$ et $\hat{\phi}_{r\beta}$ de la manière suivante :

$$\left\{ \begin{array}{l} \hat{\phi}_r = \sqrt{\hat{\phi}_{r\alpha}^2 + \hat{\phi}_{r\beta}^2} \\ \cos(\hat{\psi}) = \frac{\hat{\phi}_{r\alpha}}{\hat{\phi}_r} \\ \sin(\hat{\psi}) = \frac{\hat{\phi}_{r\beta}}{\hat{\phi}_r} \end{array} \right. \quad (3.24)$$

Les modes mécaniques sont lents par rapport aux modes électriques. Le nouveau vecteur d'état des modes électriques, le modèle linéaire variant dans le temps est caractérisé par un système d'ordre quatre représenté ci-dessous :

$$\left\{ \begin{array}{l} \frac{dX_{el}(t)}{dt} = A(\omega_m)X_{el}(t) + BU(t) \\ Y(t) = CX_{el}(t) \end{array} \right. \quad (3.25)$$

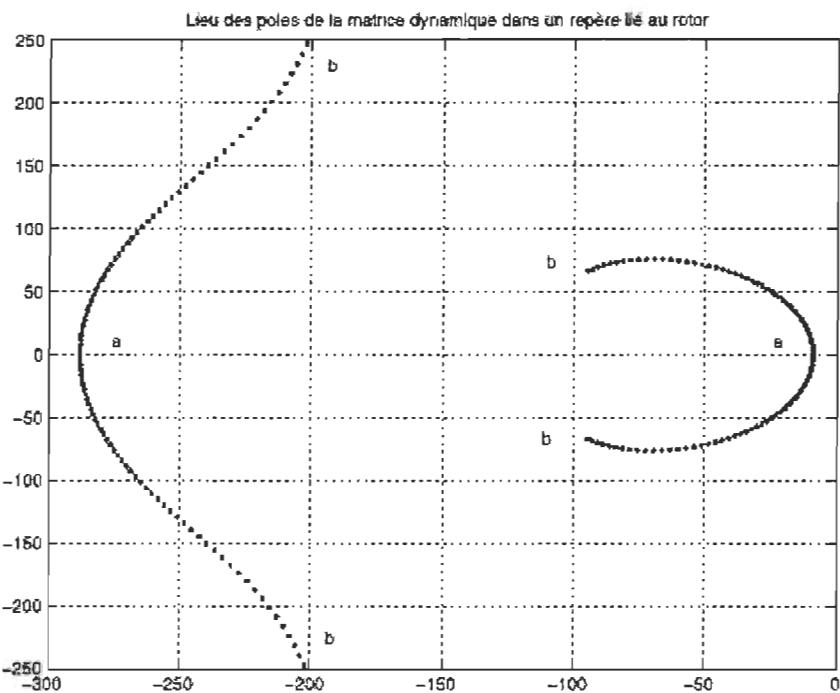


Figure 3.3: lieu des pôles dans un repère lié au stator.

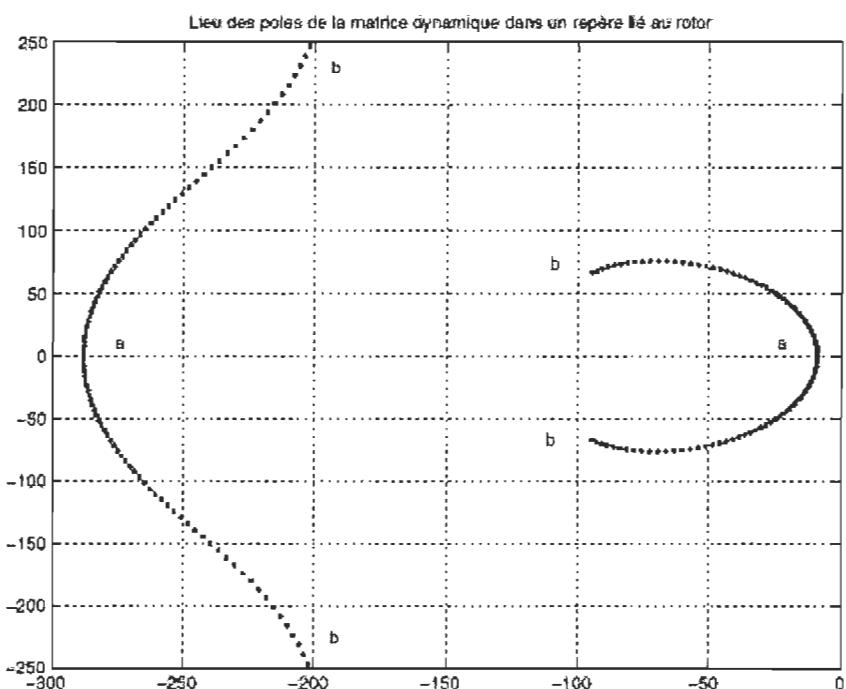


Figure 3.4: lieu des pôles dans un repère lié au rotor.

$$\begin{aligned}
A(\omega_m) = & \begin{bmatrix} - (aR_s + c \frac{M_{sr}}{L_r} Rr) & 0 & \frac{c}{L_r} Rr & c \omega_m \\ 0 & - (aR_s + c \frac{M_{sr}}{L_r} Rr) & -c \omega_m & \frac{c}{L_r} Rr \\ \frac{M_{sr}}{L_r} Rr & 0 & -\frac{1}{L_r} Rr & -\omega_m \\ 0 & \frac{M_{sr}}{L_r} Rr & \omega_m & -\frac{1}{L_r} Rr \end{bmatrix} \\
B = & \begin{bmatrix} a & 0 \\ 0 & a \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.26)
\end{aligned}$$

Le vecteur de sortie est composé des deux courants statoriques et, le vecteur de commande U est composé des deux composantes du vecteur de tensions statoriques. Le vecteur d'état est composé des courants statoriques $i_{s\alpha}$ et $i_{s\beta}$ et des composantes du flux rotorique $\phi_{r\alpha}$ et $\phi_{r\beta}$.

$$U = \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad Y = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} \quad X_{el}(t) = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Phi_{r\alpha} \\ \Phi_{r\beta} \end{bmatrix} \quad (3.27)$$

Le système d'équations ci-dessus représente le modèle classique utilisé dans la plupart des applications. Cependant, il ne permet pas de modéliser correctement le couplage des modes électriques et mécaniques et le fonctionnement sans capteur mécanique.

3.2.6.2. Choix du repère

Nous avons deux choix de repères applicables pour l'estimation du flux rotorique: le repère lié au rotor et celui lié au stator.

Le repère choisi est celui lié au stator (en raison de l'invariance dans le temps des matrices de transformation).

Discussion sur la modélisation de la machine asynchrone

Le repère statorique est le plus adéquat pour l'observation. La modélisation de la machine asynchrone dans un repère lié au flux rotorique est mieux appropriée en vue de la commande vectorielle ou de la commande par flux orienté .

Les équations d'état trouvées jusqu'à maintenant sont peu optimisées pour un processeur. En effet, il reste qu'elles soient utilisables par la suite et c'est l'objet du second paragraphe.

3.3. Méthodologie de discréétisation du modèle continu de la machine asynchrone

L'objectif de ce paragraphe est de comparer plusieurs modèles discrets de la machine asynchrone pour l'estimation d'état.

Le modèle obtenu de la machine asynchrone est un modèle d'état continu. Etant donné que les systèmes à temps réels utilisent des modèles discréétisés des processus, il est important d'obtenir l'équation d'état discréétisée du système.

Le choix de degré de développement limité de e^{AT_s} est très important lors de la méthode de discréétisation car l'ordre du développement limité et la période d'échantillonnage doivent être liés. On peut noter que plus l'ordre du développement est élevé, plus la période d'échantillonnage peut être grande. En contrepartie, l'augmentation de l'ordre du développement limité impose une grande période d'échantillonnage pour pouvoir réaliser tous les calculs numériques entre deux pas de calcul. Pour cela, il faut donc réaliser un compromis entre l'ordre du développement limité et la période d'échantillonnage. Nous avons opté pour une période d'échantillonnage de $400 \mu s$, suffisante pour exécuter les algorithmes de mesure, de commande et d'estimation sur notre DSP (carte de commande à base d'un DSP de Texas Instrument) avec un programme en langage Visual C++ (voir annexe 2).

Les différentes méthodes de discréétisation du modèle de la machine que nous avons testé sont récapitulées ci-dessous [17]:

- Modèle 1: développement limité à l'ordre 2 de e^{AT_s} .
- Modèle 2: développement limité à l'ordre 2 de e^{AT_s} et B_d à un ordre supérieur.

3.3.1 Structure des matrices A_d et B_d

Le modèle d'état discrétilisé recherché est :

$$\begin{cases} \frac{dX_{el}(t)}{dt} = A(\omega_m)X_{el}(t) + BU(t) \\ Y(t) = CX_{el}(t) \end{cases} \quad (3.28)$$

On constate que ces matrices possèdent des symétries et antisymétries. Leurs termes sont des fonctions paires ou impaires en fonction de la vitesse de rotation du moteur. La matrice A dans un repère statorique peut être écrite de manière simplifiée :

$$A = \begin{bmatrix} \alpha & 0 & \beta & c\omega_m \\ 0 & \alpha & -c\omega_m & \beta \\ \gamma & 0 & \delta & -\omega_m \\ 0 & \gamma & \omega_m & \delta \end{bmatrix} \text{ avec } \begin{cases} \alpha = - (aR_s + c \frac{M_{sr}}{L_r} Rr) \\ \beta = \frac{c}{L_r} Rr \\ \gamma = \frac{M_{sr}}{L_r} Rr \\ \delta = \frac{-1}{L_r} Rr \end{cases} \quad (3.29)$$

Le calcul de A^2 et A^3 montre que ce sont des matrices possédant les mêmes symétries et antisymétries que la matrice A. Nous allons démontrer que les matrices A^n et A^{n+1} ont alors les mêmes symétries et antisymétries que le matrice A.

Démonstration: soit

$$A^n = a_{11}[n] I_{11} + b_{11}[n] J_{11} + a_{12}[n] I_{12} + b_{12}[n] J_{12} + a_{21}[n] I_{21} + b_{21}[n] J_{21} + a_{22}[n] I_{22} + b_{22}[n] J_{22} \quad (3.30)$$

Les termes en "a" sont des fonctions paires tandis que les termes en "b" sont des fonctions impaires par rapport à la vitesse de rotation ω_m avec:

$$I_{11} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad J_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad I_{12} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad J_{12} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$I_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad J_{21} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \quad I_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad J_{21} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Nous allons démontrer que A^{n+1} conserve les mêmes propriétés que la matrice A^n . Les matrices $A^0 = I$ et $A^1 = A$ peuvent s'écrire de la manière suivante : $A^0 = I = I_{11} + I_{22}$.

Avec :
$$\begin{cases} a_{11}[0] = a_{22}[0] = 1 \\ a_{12}[0] = a_{21}[0] = 0 \\ b_{11}[0] = b_{12}[0] = b_{21}[0] = b_{22}[0] = 0 \end{cases}$$

$$A = \alpha I_{11} + \beta I_{12} + c \omega_m J_{12} + \gamma I_{21} + \delta I_{22} - \omega_m J_{22}$$

avec :
$$\begin{cases} a_{11}[1] = \alpha \\ a_{12}[1] = \beta \\ a_{21}[1] = \gamma \\ a_{22}[1] = \delta \end{cases} \quad \text{et} \quad \begin{cases} b_{11}[1] = b_{21}[1] = 0 \\ b_{12}[0] = c \omega_m \\ b_{22}[1] = -\omega_m \end{cases}$$

L'expression générale donnée en équation (3.30) est donc bien valide pour $n=0$ et $n=1$.

Le calcul de A^{n+1} donne :

$$A^{n+1} = a_{11}[n+1]I_{11} + b_{11}[n+1]J_{11} + a_{12}[n+1]I_{12} + b_{12}[n+1]J_{12} + a_{21}[n+1]I_{21} + b_{21}[n+1]J_{21} + a_{22}[n+1]I_{22} + b_{22}[n+1]J_{22}$$

Avec
$$\begin{cases} a_{11}[n+1] = \alpha a_{11}[n] + \gamma a_{12}[n] \\ a_{12}[n+1] = \beta a_{11}[n] - c \omega_m b_{11}[n] + \delta a_{12}[n] + \omega_m b_{12}[n] \\ a_{21}[n+1] = \alpha a_{21}[n] + \gamma a_{22}[n] \\ a_{22}[n+1] = \beta a_{21}[n] - c \omega_m b_{21}[n] + \delta a_{22}[n] + \omega_m b_{22}[n] \end{cases}$$

et
$$\begin{cases} b_{11}[n+1] = \alpha b_{11}[n] + \gamma b_{12}[n] \\ b_{12}[n+1] = \beta b_{11}[n] + c \omega_m a_{11}[n] + \delta b_{12}[n] - \omega_m a_{12}[n] \\ b_{21}[n+1] = \alpha b_{21}[n] + \gamma b_{22}[n] \\ b_{22}[n+1] = \beta b_{21}[n] + c \omega_m a_{21}[n] + \delta a_{22}[n] - \omega_m a_{22}[n] \end{cases}$$

Les termes devant les matrices I_{ij} sont des sommes des fonctions paires tandis que les termes devant les matrices J_{ij} sont des sommes de fonctions impaires. La parité et l'imparité des coefficients des matrices A, A^2, A^3, \dots, A^n sont vérifiées quel que soit n . La matrice A_d possède donc la même structure puisque le développement limité à l'ordre n de $A_d = e^{AT_s}$ est égal à :

$$A_d = e^{AT_s} = I + AT_s + (AT_s)^2 / 2 + \dots + (AT_s)^n / n!$$

La parité ou imparité des coefficients de la matrice $A_d = e^{AT_s}$ réduisent le nombre de termes à calculer (8 termes au lieu de 16 termes): le volume de calcul réalisé par le DSP sera bien donc divisé par deux.

On peut remarquer que la matrice d'entrée B_d possède les mêmes propriétés que la matrice $A_d = e^{AT_s}$. Nous allons démontrer que certains termes sont pairs tandis que d'autres sont impairs et ceci quel que soit le développement limité de $A_d = e^{AT_s}$ à partir de la définition de B donnée par l'équation (3.31), nous pouvons écrire l'équation (3.32) en utilisant les matrices données par l'équation (3.31).

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad J_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad O_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{\sigma L s} & 0 \\ 0 & \frac{1}{\sigma L s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = a \begin{bmatrix} I_2 \\ O_2 \end{bmatrix} \quad (3.31)$$

$$\int_0^{Ts} e^{At} B dt \Rightarrow B_d = \int_0^{Ts} \sum_{n=0}^{+\infty} \left(\frac{(At)^n}{n!} \right) B dt \quad (3.32)$$

soit

$$B_d = \sum_{n=0}^{+\infty} \frac{T^{n+1}}{(n+1)!} A^n B$$

Nous utilisons alors la définition de la matrice A^n donnée par l'équation (3.30), ceci nous donne :

$$\text{soit } B_d = a \sum_{n=0}^{+\infty} \frac{T^{n+1}}{(n+1)!} \begin{bmatrix} a_{11}[n] & b_{11}[n] \\ -b_{11}[n] & a_{11}[n] \\ a_{21}[n] & b_{21}[n] \\ -b_{21}[n] & a_{21}[n] \end{bmatrix}$$

donc :

$$B_d = a_1 \begin{bmatrix} I_2 \\ O_2 \end{bmatrix} + b_1 \begin{bmatrix} J_2 \\ O_2 \end{bmatrix} + a_2 \begin{bmatrix} O_2 \\ I_2 \end{bmatrix} + b_2 \begin{bmatrix} O_2 \\ J_2 \end{bmatrix}$$

$$\text{avec } \begin{cases} a_1 = a \sum_{n=0}^{+\infty} \frac{Ts^{n+1}}{(n+1)!} a_{11}[n] \\ b_1 = a \sum_{n=0}^{+\infty} \frac{Ts^{n+1}}{(n+1)!} b_{11}[n] \\ a_2 = a \sum_{n=0}^{+\infty} \frac{Ts^{n+1}}{(n+1)!} a_{21}[n] \\ b_2 = a \sum_{n=0}^{+\infty} \frac{Ts^{n+1}}{(n+1)!} b_{21}[n] \end{cases}$$

Pour conclure, on peut dire que les termes symétriques et antisymétriques sont respectivement des fonctions paires et impaires par rapport à la vitesse de rotation du rotor. Maintenant, nous allons établir les équations des trois modèles discrets de la machine asynchrone dans le repère fixe $\alpha - \beta$.

Les coefficients a_1 et a_2 sont pairs et les coefficients b_1 et b_2 sont impairs. Nous remarquons alors que les termes pairs sont en relation avec les matrices I_2 tandis que les termes impairs sont en relation avec les matrices J_2 .

3.3.2. Modèle 1 : Développement limité à l'ordre 2 de e^{ATs} : Calcul de A_d et B_d :

Les matrices A_d et B_d sont calculées par un développement limité à l'ordre 2 de e^{ATs} avec une période d'échantillonnage de 400 μ s. Ainsi,

$$e^{ATs} \approx A_d = I + ATs + (ATs)^2 / 2 \quad (3.38)$$

et la matrice B_d se calcule par :

$$\int_0^{Ts} e^{At} B dt = A^{-1} (e^{ATs} - I) B \approx B_d = Ts (I + \frac{ATs}{2}) B \quad (3.39)$$

L'utilisation de la forme simplifiée de l'équation (3.29) permet le calcul des matrices A_d et B_d :

$$A_d = \begin{bmatrix} a_{11} & b_{11} & a_{12} & b_{12} \\ -b_{11} & a_{11} & -b_{12} & a_{12} \\ a_{21} & b_{21} & a_{22} & b_{22} \\ -b_{21} & a_{21} & -b_{22} & a_{22} \end{bmatrix} \quad B_d = \begin{bmatrix} a_1 & 0 \\ 0 & a_1 \\ a_2 & 0 \\ 0 & a_2 \end{bmatrix} \quad (3.40)$$

et $C_d = C$

$$\text{avec } \left\{ \begin{array}{l} a_{11} = 1 + \alpha Ts + (\alpha^2 + B\gamma)T^2 s / 2 \\ a_{12} = \beta Ts (1 + (\alpha + \delta) \frac{Ts}{2}) + c\omega_m^2 T^2 s / 2 \\ a_{21} = \gamma Ts (1 + (\alpha + \delta) Ts / 2) \\ a_{22} = 1 + \delta Ts + (\delta^2 + B\gamma)T^2 s / 2 - \omega_m^2 T^2 s / 2 \\ b_{11} = c\omega_m^2 T^2 s / 2 = -cb_{21} \\ b_{12} = (cTs (1 + (\alpha + \delta) Ts / 2) - \beta T^2 s / 2) \omega_m \\ b_{21} = -\gamma \omega_m^2 T^2 s / 2 \\ b_{22} = (-Ts + (c\gamma - 2\delta) T^2 s) \omega_m \\ a_1 = \alpha Ts (1 + \alpha Ts / 2) \\ a_2 = \alpha \gamma \frac{Ts^2}{2} \end{array} \right.$$

La structure matricielle présente une symétrie et antisymétrie qui vont permettre par la suite une simplification de l'algorithme et la mémoire de DSP.

La complexité des calculs peut être évaluée en comptant le nombre de multiplications et d'additions que doit exécuter le DSP ou le microcontrôleur. Les résultats sont donnés dans le tableau (3.1) qui permet de comparer les coefficients des matrices A_d et B_d calculés par cette approximation aux coefficients "vrais" des matrices A_d et B_d exactes.

L'erreur absolue maximale est calculée de la manière suivante :

$$\text{Erreur absolue maximale} = \text{Max} \left| \text{coefficient exact}(\omega_m) - \text{coefficient approché}(\omega_m) \right|$$

Les tableaux (3.1) et (3.2) récapitulent les erreurs absolues maximales commises sur les coefficients des matrices A_d et B_d pour la machine asynchrone (voir annexe 2). Les résultats obtenus montrent que l'erreur absolue la plus importante sur les coefficients de A_d est d'environ $1.8 \cdot 10^{-2}$. Ces résultats paraissent pour le moment peu acceptables pour l'observation d'état. De plus, il y a des erreurs sur les coefficients de la matrice B_d . La matrice B_d ne modélisera pas correctement le "comportement" de la machine asynchrone en entrée. Ce phénomène est encore plus important lorsque les tensions $V_{s\alpha}$ et $V_{s\beta}$ réellement appliquées sur la machine sont mal connues.

Tableau 3.1: Nombre d'opérations pour le calcul de A_d et B_d dans le cas du modèle 1.

Matrice	Nombre de multiplication	Nombre d'addition
A_d	8	2
B_d	0	0

3.3.3. Modèle 2: Calcul de A_d par un développement limité à l'ordre 2 de e^{At_s} et calcul de B_d à un ordre supérieur

La matrice A_d est une fonction du second ordre dans le modèle 1 par rapport à la vitesse de rotation du moteur tandis que les termes de la matrice B_d restent constants. Nous recherchons alors des fonctions du second ordre pour A_d et du premier ordre pour B_d telles que :

Tableau 3.2: Nombre d'opérations pour le calcul de A_d et B_d dans le cas du modèle 2

Matrice	Nombre de multiplication	Nombre d'addition
A_d	8	2
B_d	3	1

$$\begin{cases} A_d = f(\omega_m, \omega^2_m) \\ B_d = g(\omega_m) \end{cases}$$

On peut calculer la matrice A_d par un développement limité à l'ordre 2 de e^{ATs} et B_d par un développement limite à l'ordre 3 de e^{ATs} . Ainsi :

$$\begin{cases} A_d = I + ATs + \frac{(ATs)^2}{2} \\ B_d = Ts(I + \frac{(ATs)}{2} + \frac{(ATs)^3}{3!})B \end{cases} \quad (3.41)$$

La matrice A_d reste inchangée par rapport au modèle 1 puisque le développement de e^{ATs} est le même, mais maintenant les coefficients de la matrice B_d sont fonction de la vitesse de rotation ω_m . La matrice B_d est définie par l'équation (3.42) :

$$B_d = \begin{bmatrix} a_1 & b_1 \\ -b_1 & a_1 \\ a_2 & b_2 \\ -b_2 & a_2 \end{bmatrix} \quad \text{avec :} \quad \begin{cases} a_1 = aTs(1 + \alpha Ts/2 + (\alpha^2 + \beta\gamma)T^2s/6) \\ b_1 = ac\gamma\omega_m T^3s/6 \\ a_2 = aT^2s(\gamma/2 + (\alpha + \delta)\gamma Ts/6) \\ b_2 = -a\gamma\omega_m T^3s/6 \end{cases} \quad (3.42)$$

On compare les coefficients de la matrice B_d calculés par cette nouvelle approximation aux vrais coefficients de la matrice B_d . Le tableau (3.3) récapitule les erreurs absolues maximales commises sur ces coefficients.

Les résultats obtenus montrent que les erreurs absolues maximales de la matrice B_d du modèle 2 sont bien inférieures aux erreurs absolues de la matrice B_d du modèle 1. Ceci est évident puisque la matrice B_d est maintenant fonction de la vitesse de rotation du

rotor. La complexité des calculs est évaluée en comptant le nombre de multiplications et d'additions. Les résultats sont données dans le tableau (3.3).

Tableau 3.3: Erreurs absolues maximales sur les coefficients de B_d

coefficient	Modèle 1	Modèle 2
a_1	$3.55 \cdot 10^{-5}$	$1.12 \cdot 10^{-6}$
b_1	$1.36 \cdot 10^{-5}$	$8.36 \cdot 10^{-7}$
a_2	$3.28 \cdot 10^{-7}$	$9.20 \cdot 10^{-9}$
b_{21}	$3.32 \cdot 10^{-7}$	$1.42 \cdot 10^{-8}$

Tableau 3.3: Erreurs absolues maximales sur les coefficients de A_d

coefficient	Modèle 1	Modèle 2
a_{11}	$2.42 \cdot 10^{-4}$	$2.42 \cdot 10^{-4}$
b_{11}	$2.00 \cdot 10^{-4}$	$2.00 \cdot 10^{-4}$
a_{12}	$1.80 \cdot 10^{-2}$	$1.80 \cdot 10^{-2}$
b_{12}	$4.64 \cdot 10^{-3}$	$4.64 \cdot 10^{-3}$
a_{21}	$3.20 \cdot 10^{-6}$	$3.20 \cdot 10^{-6}$
b_{21}	$3.39 \cdot 10^{-6}$	$3.39 \cdot 10^{-6}$
a_{22}	$2.59 \cdot 10^{-4}$	$2.59 \cdot 10^{-4}$
b_{22}	$2.03 \cdot 10^{-4}$	$2.03 \cdot 10^{-4}$

En conclusion, les erreurs absolues commises lors de la discréttisation sont bien inférieures à celles du modèle 1 pour un coût algorithmique supplémentaire, seulement deux multiplications et une addition.

3.4. Augmentation de l'état :

Nous allons étendre le modèle mathématique de la machine asynchrone pour l'estimation de la vitesse comme on l'a déjà fait en (3.3.2).

Le modèle d'état discret dans le repère $\alpha - \beta$ devient $X_a = [i_{s\alpha} \quad i_{s\beta} \quad \phi_{r\alpha} \quad \phi_{r\beta} \quad \omega_m]$

3.4.1. Linéarisation du modèle :

Ce modèle d'état étendu car la vitesse apparaît dans le vecteur d'état et dans la matrice de transition .

3.4.1.1. Calcul de A_d et B_d par un développement limite d'ordre 1 de e^{ATs}

Les matrices A_d et B_d sont calculées par un développement limite à l'ordre 1 de e^{ATs} avec une période d'échantillonnage de $400 \mu s$. Ainsi,

$$e^{ATs} \equiv A_d = I + ATs . \quad (3.43)$$

Et la matrice B_d se calcule par :

$$\int_0^{Ts} e^{At} B dt = A^{-1} (e^{ATs} - I) B \approx B_d = TsB . \quad (3.44)$$

Les matrices d'entrées et de sorties continues du système sont notées A,B et C et celles d'entrées et des sorties discrètes sont A_d , B_d et C_d .

On assume que le temps est très court comparé à la dynamique du système.

Le calcul des matrices A_d et B_d en utilisant la forme simplifiée de l'équation (3.29) donne :

$$A_d = \begin{bmatrix} a_{11} & 0 & a_{13} & a_{14} & 0 \\ 0 & a_{11} & -a_{14} & a_{13} & 0 \\ a_{31} & 0 & a_{33} & a_{34} & 0 \\ 0 & a_{31} & -a_{34} & a_{33} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad B_d = \begin{bmatrix} a_1 & 0 \\ 0 & a_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad C_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

donc

$$\left\{ \begin{array}{l} a_{11} = 1 + \alpha Ts \\ a_{13} = \beta Ts \\ a_{31} = \gamma Ts \\ a_{33} = 1 + \delta Ts \\ a_{14} = c \omega_m \\ a_{34} = (-Ts) \omega_m \\ a_1 = aTs \end{array} \right. \quad (3.45)$$

3.4.1.2. Simulation du filtre de Kalman d'ordre 1 de e^{ATs}

Le modèle d'état discréteisé est caractérisé par:

$$\left\{ \begin{array}{l} \phi = A_d(\omega)X(k) + B_dU(k) \\ Y(k) = CX(k) \end{array} \right. \quad (3.46)$$

$$h = cx = \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}$$

$$\frac{\partial \phi}{\partial x} = \begin{bmatrix} 1 + Ts * a_{11} & 0 & Ts * a_{13} & Ts * a_{14} & Ts * c_1 \\ 0 & 1 + Ts * a_{11} & -Ts * a_{14} & Ts * a_{13} & Ts * c_2 \\ Ts * a_{31} & 0 & 1 + Ts * a_{33} & Ts * a_{34} & Ts * c_3 \\ 0 & Ts * a_{31} & -Ts * a_{34} & 1 + Ts * a_{33} & Ts * c_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.47)$$

$$\frac{\partial h}{\partial x} = C_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Maintenant, nous avons la forme discrète du modèle avec laquelle on peut calculer les matrices et les vecteurs qui seront nécessaires pour la simulation et l'implantation avec Matlab/Simulink.

La réalisation du filtre de Kalman avec Simulink (figure 3.5) seul est trop compliquée mais, elle devient facile avec le programme Matlab S_function. L'autre avantage est que la conversion en programme assembleur est facile .

sous_ système est inséré dans le système qui contient le filtre de Kalman qui est dans le système appelé programme1. Les sorties du système sont les variables d'état du flux rotorique, le courant statorique et la vitesse rotorique. Les entrée mesurées sont le courant et la tension statorique. Ce modèle se base sur celui de Peter vas [17]

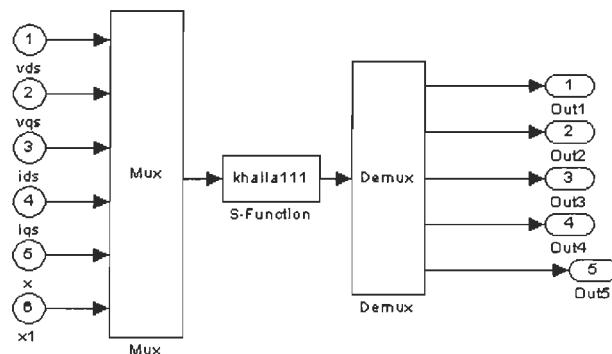


Figure 3.5 :Modèle Simulink du filtre de Kalman

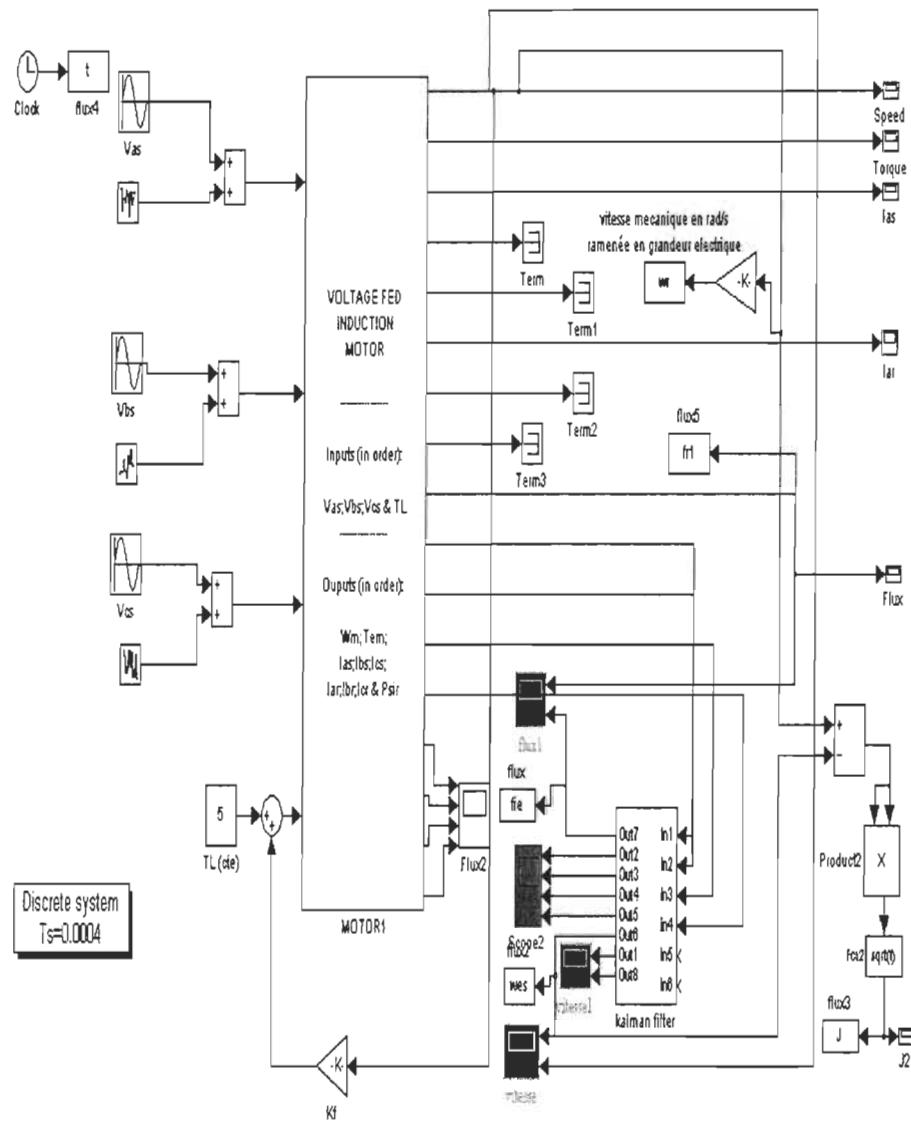


Figure 3.6 : Modèle Simulink (machine asynchrone et le filtre de Kalman)

3.4.1.2.1. Résultats de la simulation :

Les programmes et les figures (3.7 et 3.8) montrent respectivement les vitesses et les flux estimés et réels.

La réponse du filtre de Kalman est peu différente en transitoire de celle de référence.

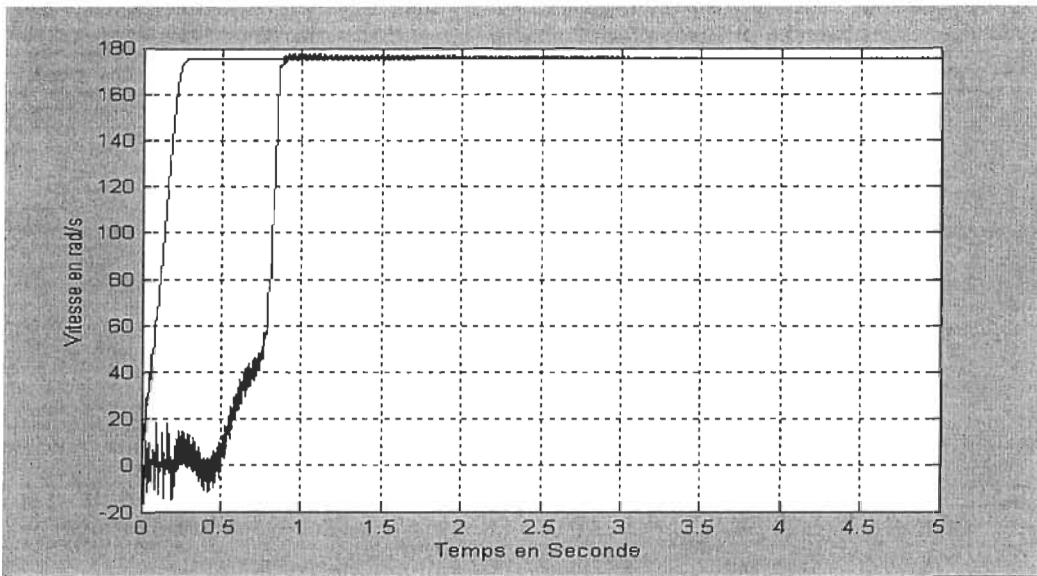


Figure 3.7 :Flux réel comparé au Flux estimé

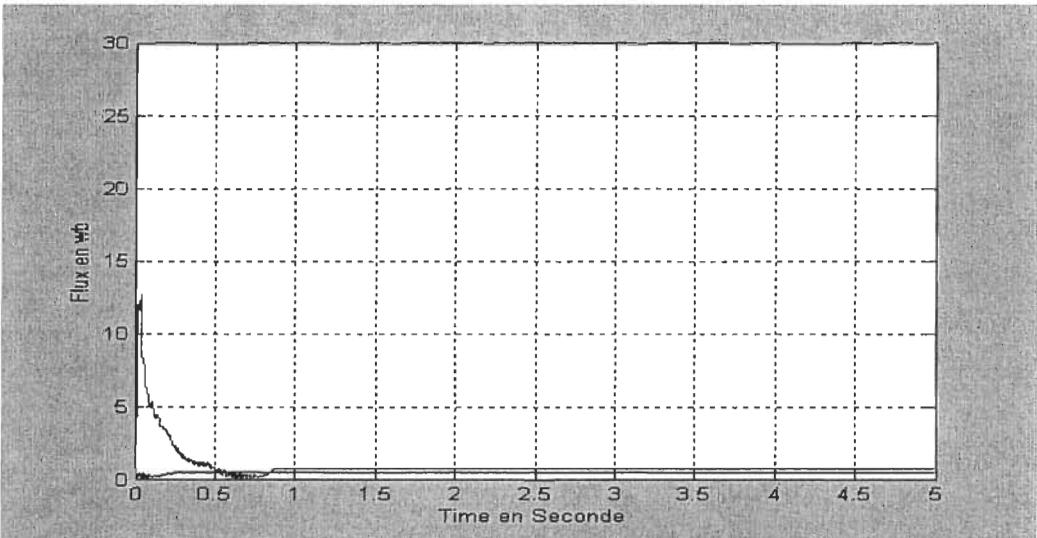


Figure 3.8 :vitesse réelle comparée à la vitesse estimée

Discussion :

Le modèle de ce système permet la comparaison des résultats en-ligne. En effet le filtre de Kalman a été intégré à la simulation sous forme de S-function Matlab.

L'algorithme du filtre de Kalman classique d'ordre 1 aboutit à des résultats de simulation acceptables mais, nécessite un DSP performant à l'étape de l'implantation. En effet, si on tient compte de garder la taille du problème raisonnable et obtenir plus de performance avec le minimum d'instruction (DSP raisonnable), on serait obligé d'effectuer le même

travail mais avec un filtre de Kalman optimisé d'ordre 2 et un logiciel plus performant comme le visual C++ (compilation rapide avec l'intégration d'Euler).

3.4.1.2.2. Calcul de A_d et B_d par un développement limite d'ordre 2 de e^{ATs}

Le système d'équations d'état devient en se basant sur (3.40):

$$X_a[k+1] = \begin{bmatrix} a_{11} & b_{11} & a_{12} & b_{12} & 0 \\ -b_{11} & a_{11} & -b_{12} & a_{12} & 0 \\ a_{21} & b_{21} & a_{22} & b_{22} & 0 \\ -b_{21} & a_{12} & -b_{12} & a_{12} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Phi_{r\alpha} \\ \Phi_{r\beta} \\ \omega_m \end{bmatrix} + \begin{bmatrix} a_1 & 0 \\ 0 & a_1 \\ a_2 & 0 \\ 0 & a_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{s\alpha} \\ V_{s\beta} \end{bmatrix} \quad (3.48)$$

La matrice de transition :

$$Y[k+1] = HX_a = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \\ \Phi_{r\alpha} \\ \Phi_{r\beta} \\ \omega_m \end{bmatrix} \quad (3.49)$$

3.4.1.2.3. Calcul des gains de correction

Le calcul des gains de correction d'un observateur linéaire étendu *OLE* est effectué à partir des valeurs propres de la matrice de transition F de l'observation linéarisée autour de la trajectoire estimée.

$$F[k] = \left[\frac{\partial X_a}{\partial X} \right] = \begin{bmatrix} a_{11} & b_{11} & a_{12} & b_{12} & f_1 \\ -b_{11} & a_{11} & -b_{12} & a_{12} & f_2 \\ a_{21} & b_{21} & a_{22} & b_{22} & f_3 \\ -b_{21} & a_{12} & -b_{12} & a_{12} & f_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ avec : } \begin{cases} C_1 = \frac{\partial i_{s\alpha}[k+1]}{\partial \omega_m} \\ C_2 = \frac{\partial i_{s\beta}[k+1]}{\partial \omega_m} \\ C_3 = \frac{\partial \phi_{r\alpha}[k+1]}{\partial \omega_m} \\ C_4 = \frac{\partial \phi_{r\beta}[k+1]}{\partial \omega_m} \end{cases} \quad (3.50)$$

$$\left\{
\begin{aligned}
C1 &= 0.5 * c_K * sn_K * Ts * Ts * x(2) + c_K * Ts * Ts * x(5) * x(3) + \\
&\quad (c_K * Ts * (1 + (alpha + gama_K) * Ts/2) - b_K * Ts * Ts/2) * x(4), \\
C2 &= -0.5 * c_K * sn_K * Ts * Ts * x(1) + c_K * Ts * Ts * x(5) * x(4) - \\
&\quad c_K * Ts * (1 + (alpha_K + gama_K) * Ts/2) - b_K * Ts * Ts/2 \\
C3 &= -0.5 * sn_K * Ts * Ts * x(2) - Ts * Ts * x(5) * x(3) + \\
&\quad (0.5 * (c_K * sn_K - 2 * gama_K) * Ts * Ts - Ts) * x(4), \\
C4 &= 0.5 * sn_K * Ts * Ts * x(1) - Ts * Ts * x(5) * x(4) - \\
&\quad (0.5 * (c_K * sn_K - 2 * gama_K) * Ts * Ts - Ts) * x(3),
\end{aligned}
\right. \quad (3.51)$$

Où les coefficients $a_{11}, b_{11}, a_{12}, b_{12}, a_{22}, b_{22}, a_1, a_2$ sont déjà précisés à l'expression (3.40).

L'erreur de reconstruction $\tilde{X}_a = X[K] - \tilde{X}_a[K]$ est régie par l'équation [18] :

$$\tilde{X}_a[K+1] = (I - L[K+1]H[K+1])F[K+1]\tilde{X}_a[K]$$

Conclusion :

L'étude de la sensibilité des trois modèles discrets en fonction de la précision de la méthode de discréttisation montre que ces trois modèles sont quasi-équivalents et, qu'il est nécessaire de travailler sur la désensibilisation de l'estimateur aux variations de paramètres plutôt qu'à la désensibilisation aux effets de la discréttisation.

Le réglage et l'algorithme "efficace" d'un filtre de Kalman optimal pour l'estimation du flux rotorique et des courants statoriques sera décrit au chapitre suivant, nous allons effectué un rappel concernant les reconstructeurs d'état et leurs moyens de réglage.

Chapitre 4

Conception de l'algorithme du filtre de Kalman

Résumé : Après avoir effectué la modélisation, et la discréétisation du filtre de Kalman, ce chapitre propose une solution optimale conçue à cet effet et paramétrée par la manipulation matricielle.

Les résultats ainsi obtenus sont comparés à ceux de l'observateur de LMI. Enfin, pour une validation totale de l'observateur, ce chapitre s'achèvera sur une analyse en simulation de sa robustesse paramétrique.

4.1. Introduction

Dans les chapitres 2 et 3, on a parlé de l'estimateur en particulier, l'estimateur de Kalman, dont on a introduit les concepts, les principes généraux du filtre de Kalman. Nous sommes rendus à la fin de ce mémoire et il serait utile de vérifier à l'aide de la simulation quelques résultats d'estimation. Nous abordons donc, dans ce chapitre la conception de l'algorithme de Kalman optimal, les résultats obtenus seront interprétées et comparées à ceux de l'observateur de LMI.

4.2. Choix des matrices Q et R

4.2.1. Pré-réglage automatique du filtre de Kalman

Le réglage du filtre de Kalman par choix de Q et R se fait d'une façon empirique jusqu'à l'obtention d'un comportement optimal du filtre. Cette méthode ne garantit pas la bonne robustesse et la bonne convergence du filtre face aux variations paramétriques et aux perturbations. Les matrices d'état des variances-covariances d'erreur renseignent sur la variance d'état estimée, donc sur la qualité de l'estimateur et sa précision. On peut conclure que le bon choix de Q et R est un bon choix de l'estimateur.

4.2.1.1. Structure et réglage de R :

Le filtre de Kalman utilise les courants statoriques $i_{s\alpha}$ et $i_{s\beta}$ comme entrées mesurables :

$$R = \begin{bmatrix} \sigma^2 & \rho\sigma^2 \\ \rho\sigma^2 & \sigma^2 \end{bmatrix} \quad \text{avec } \rho = \frac{E(Per_{s\alpha} Per_{s\beta})}{\sqrt{(Per_{s\alpha}^2)(Per_{s\beta}^2)}} \quad (4.1)$$

où $Per_{is\alpha}$ et $Per_{is\beta}$ représentent les perturbations sur la mesure des courants.

Le passage d'un système triphasé à un système diphasé se fait par T_{32} :

$$\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = T_{32}(i_s + Per_{is}) \quad \text{avec} \quad i_s = \begin{bmatrix} i_{s1} \\ i_{s2} \\ i_{s3} \end{bmatrix} \quad \text{et} \quad Per_{is} = \begin{bmatrix} Per_{is1} \\ Per_{is2} \\ Per_{is3} \end{bmatrix}$$

Per_{is} : Représente un bruit blanc de moyenne nulle. L'espérance mathématique est :

$$E \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = T_{32} E(i_s) = T_{32} i_s$$

La variance mathématique est :

$$V \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} = E((\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} - E \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix})(\begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix} - E \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix})^t)$$

$$\text{Soit } R = E((T_{32} Per_{is})(T_{32} Per_{is})^t) = T_{32} E((Per_{is})(Per_{is})^t) T_{32}^t = \sigma^2 T_{32} T_{32}^t \quad (4.2)$$

$$T_{32} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \text{ ce qui donne la matrice } R = \sigma^2 \begin{bmatrix} \frac{3}{2} & 0 \\ 0 & \frac{3}{2} \end{bmatrix} = \frac{3}{2} \sigma^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Cette façon se base sur la loi des nœuds $\sum_{j=1}^3 i_{sj} = 0$; dans notre cas $\sum_{j=1}^2 i_{sj} = 0$ ce qui implique :

$$T_{32} = \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 \\ \frac{2}{\sqrt{2}} & \sqrt{2} \end{bmatrix}, \text{ notre matrice } R \text{ devient carré : } R = \sigma^2 \begin{bmatrix} \frac{3}{2} & \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{3}{2} \end{bmatrix} = \frac{3}{2} \sigma^2 \begin{bmatrix} 1 & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} & 1 \end{bmatrix}$$

La structure diagonale de R permet la conservation structurelle des matrices de variance-covariance. Donc, notre matrice doit être choisie de la façon suivante :

$$R = \frac{3}{2} \sigma^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \sigma^2_{is\alpha} & 0 \\ 0 & \sigma^2_{is\beta} \end{bmatrix} \quad (4.3)$$

4.2.1.2. Structure et réglage de Q :

La matrice Q caractérise les bruits d'état du processus (systématique). Dans notre cas la matrice Q est choisie diagonale en absence de connaissance sur le bruit du système.

$$Q = \begin{bmatrix} \sigma^2_{i_s} & 0 & 0 & 0 \\ 0 & \sigma^2_{i_s} & 0 & 0 \\ 0 & 0 & \sigma^2_{\phi_r} & 0 \\ 0 & 0 & 0 & \sigma^2_{\phi_r} \end{bmatrix} \quad (4.4)$$

La 1^{ère} matrice Q caractérise les bruits d'état du processus sur le courant i_s .

La 2^{eme} matrice Q caractérise les bruits d'état du processus sur le flux ϕ_r .

4.2.1.3. Interprétation des matrices et leur réglage de Q :

Soit R est choisie diagonale et fixe et, si on diminue $\sigma^2_{i_s}$, alors une prédiction est réalisée avec une faible incertitude. Une meilleure estimation est le résultat d'une meilleure prédiction et non de la mesure du courant statorique.

Au cas contraire : Si on diminue $\sigma^2_{i_s}$ alors une prédiction est réalisée avec plus d'incertitude ce qui entraîne une augmentation de gain de Kalman pour améliorer l'estimation.

Le rapport Q/R est important quand la matrice R est diagonale et une multiplication des deux variance-covariance par le même scalaire ne modifie pas le gain de Kalman donc notre variance-covariance possède seulement deux degrés de liberté ($\alpha_1; \alpha_2$) à ajuster successivement :

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{et} \quad Q = \begin{bmatrix} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_1 & 0 & 0 \\ 0 & 0 & \alpha_2 & 0 \\ 0 & 0 & 0 & \alpha_2 \end{bmatrix} \quad (4.5)$$

Les paramètres doivent être choisis de façon à suivre respectivement la variation de l'état et de la bande passante pour rendre l'estimateur proche de celui réellement conçu et dont le rapport des caractéristiques prix/qualité est meilleur à celui obtenu avec un capteur.

4.2.1.3.1. Pré-réglage des degrés de liberté :

Des études de l'ensemble des imperfections liées au modèle discret de la machine asynchrone permettent d'effectuer un réglage quasi-optimal du filtre. Les principales sources d'erreur de modélisation sont par ordre d'importance :

1. l'erreur d'estimation des paramètres,
2. l'erreur d'estimation de la tension d'alimentation de l'onduleur,
3. l'erreur due aux variations de la vitesse sur une période d'échantillonnage,
4. l'erreur due à une non prise en compte des pertes fer,
5. l'erreur due aux temps morts de l'onduleur,
6. l'alimentation par onduleur à MLI,
7. la troncature du développement de l'exponentielle de matrice

Le réglage effectué des coefficients (α_1, α_2) pour mieux estimer l'amplitude du flux rotorique, lorsque la valeur de la résistance rotorique est sous-estimée de 60%, est constitué de divers modes de fonctionnements .

1. l'installation du flux dans la machine,
2. le démarrage du moteur jusqu'à la vitesse nominale,
3. le fonctionnement à vitesse nominale sans couple résistant,
4. le fonctionnement à vitesse nominale avec couple résistant (2 N.m),
5. le fonctionnement à vitesse nulle avec couple résistant (2 N.m).

Le filtre de Kalman d'ordre 4 a été programmé sous Matlab. Il utilise la fonction fmins afin de rechercher le couple (α_1, α_2) le plus approprié pour minimiser le critère déterministe suivant :

$$J = \sqrt{\frac{1}{N} \sum_{K=1}^N (\phi_r[K] - \hat{\phi}_r[K])^2} \quad (4.6)$$

La solution obtenue est :

$$\begin{aligned} \alpha_1 &= 4.10^2 \\ \alpha_2 &= 6.10^{-2} H^2 \end{aligned}$$

Remarque : Le coefficient α_2 représentant le rapport entre la variance sur les composantes du flux rotorique et la variance sur les composantes du courant statorique. La dimension physique de α_2 est le Henry.

Le vecteur d'état $\hat{X}_{el}[0/0]$ et la matrice de variance-covariance des erreurs d'estimations sont initialisés à zéro. En l'absence d'information exacte sur l'état magnétique de la machine au démarrage, il est légitime de supposer les états nuls à l'instant initial. En réalité, les composantes du flux observées devraient être égales en toute rigueur au flux rémanent dans la machine.

4.2.1.3.2. Simulation du filtre de Kalman d'ordre 2 de e^{Ats} comparé à LMI

À la période d'échantillonnage de $400 \mu s$, on peut faire la compilation, la simulation en visual C++ (programme fourni en annexe 1) puis la visualisation des courbes avec Matlab (figure :4.1) des deux flux rotoriques réels.

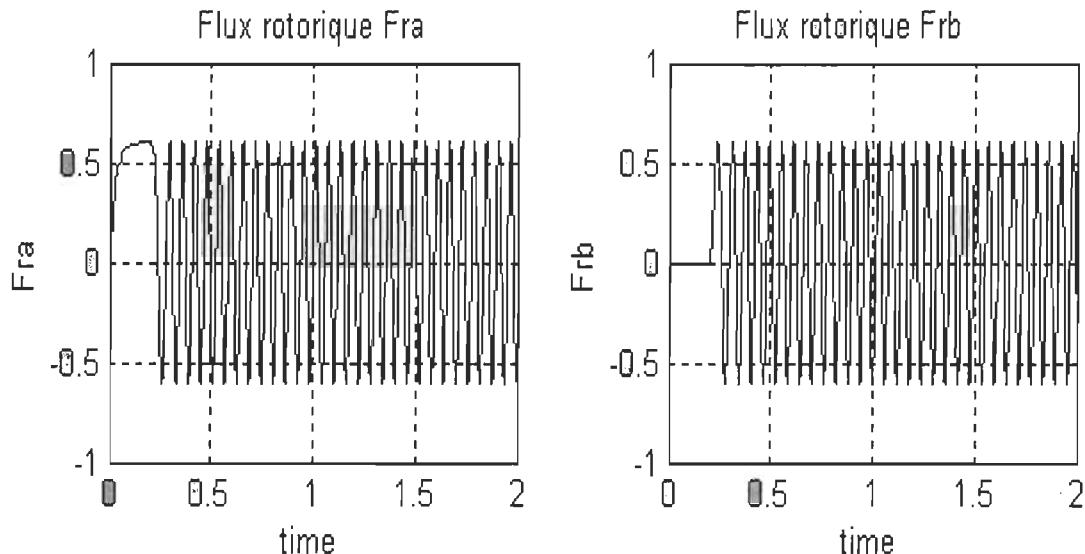


figure 4.1 : Flux rotorique réel

Les résultats de simulation de flux rotorique de filtrage de Kalman ainsi obtenus sont comparés à ceux de LMI dont la structure Simulink de l'Observateur de flux est la suivante:

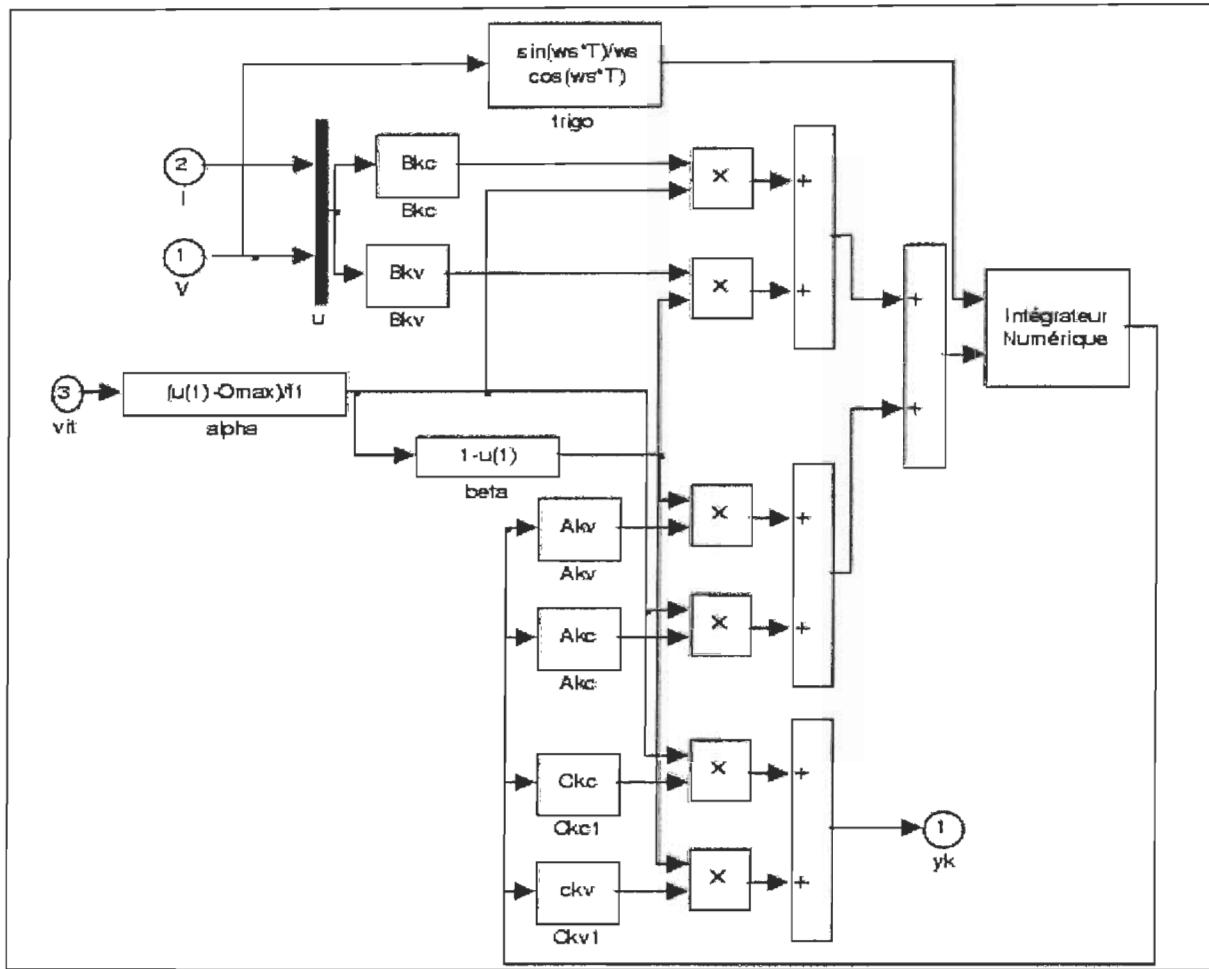


Figure 4.2: Structure Simulink de l'Observateur de flux LMI LPV

4.2.1.3.3. Analyse des performances du filtre

L'observateur de Kalman est simulé en boucle ouverte, c'est à dire que les informations disponibles en sortie de l'observateur ne sont pas utilisées pour comparer leurs performances. Les résultats de la comparaison sont donnés dans la figure 4.3. Les simulations commencent par un démarrage direct du moteur à partir d'une source de tension triphasée (220V, 60Hz). Les observateurs sont déclenchés à l'instant t égal à 0.3 s à partir de conditions initiales nulles. Cette procédure permet de tester leur comportement dynamique (oscillations, traînage, ...).

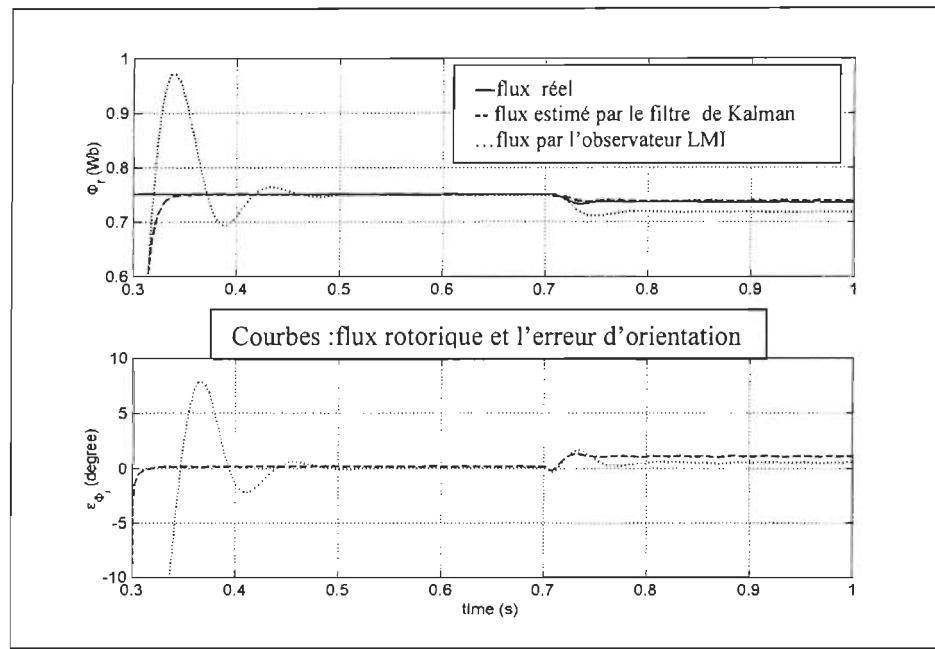
La figure (4.3a) illustre la réponse de la simulation où la valeur de la résistance rotorique est sous-estimée de 50%. On constate que le module du flux estimé par le filtre de Kalman est presque par tout semblable au flux réel par rapport à celui de l'observateur LMI.

Le réglage du filtre de Kalman ayant été réalisé dans des conditions très similaires $\Delta R_r / R_r = 60\%$, il est logique de constater un comportement “optimal” dans de telles conditions. En contre-partie, cette performance induit une plus grande erreur d'estimation de l'orientation du vecteur flux rotorique.

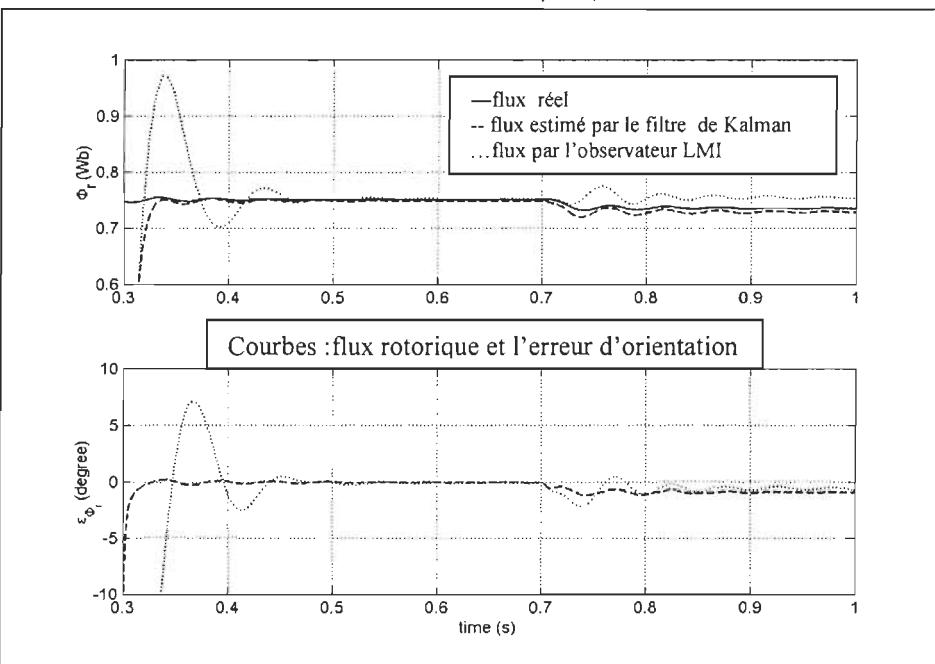
La figure (4.3b) représente une simulation où l'on surestime la valeur de la résistance rotorique de 50%. On constate également que l'erreur d'estimation de l'amplitude du flux rotorique estimée par l'observateur étalon LMI est plus grande que celle liée au filtre de Kalman. Les erreurs d'estimation en orientation du flux rotorique sont équivalentes pour les deux observateurs et restent assez faibles en régime permanent (inférieures à un degré).

Les figure (4.3c)et (4.3d) sont deux simulations respectivement une sous-estimation et une surestimation de 25 % de la résistance statorique. Les deux observateurs ont une erreur d'estimation de flux rotorique presque équivalente, et ils ne sont pas sensibles aux incertitudes de résistance statorique à la différence des observateurs précédemment proposés.

L'étude de la robustesse aux variations des autres paramètres (résistance statorique et inductance mutuelle) montre que les deux observateurs sont équivalents. Toutefois, ils présentent des comportements dynamiques très différents. L'observateur LMI a un comportement du type oscillatoire et le filtre de Kalman a un comportement du type amorti.

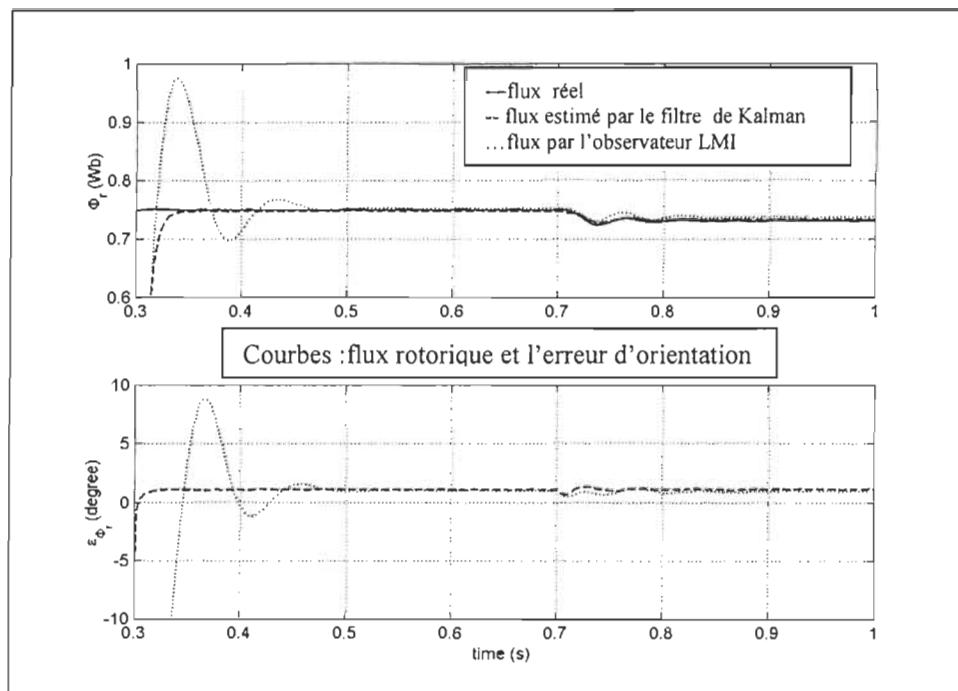


(a) estimation avec $\Delta R_r / R_r = +50\%$

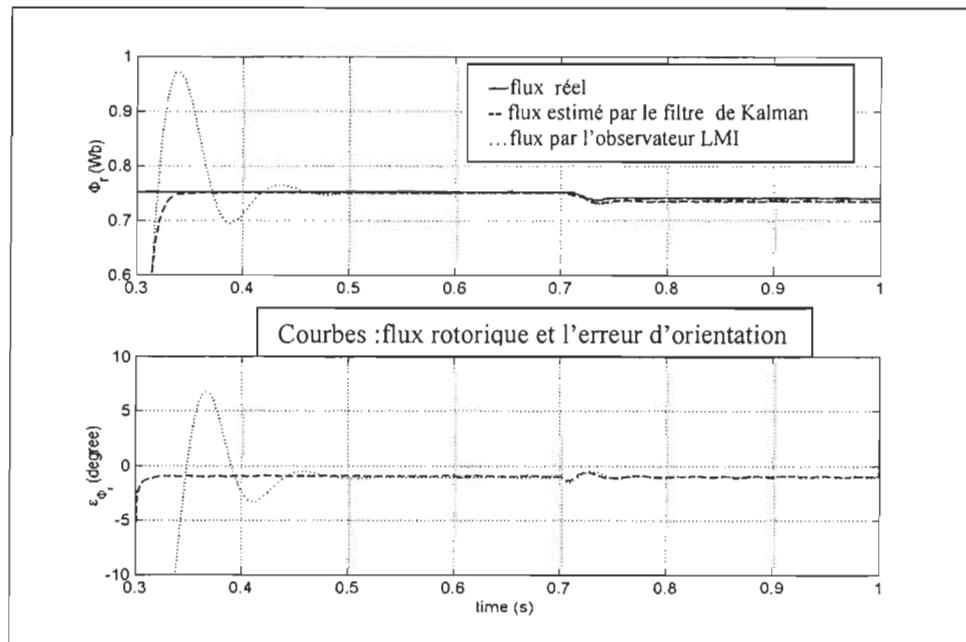


(b) estimation avec $\Delta R_r / R_r = -50\%$

Figure 4.3 : Comparaison des performances des observateurs Kalman et LMI
N.B : ε_{ϕ_r} : Erreur d'orientation



(c) estimation avec $\Delta R_r / R_r = 0\%$ et $\Delta R_s / R_s = +25\%$



(d) estimation avec $\Delta R_r / R_r = 0\%$ et $\Delta R_s / R_s = -25\%$

Figure 4.3 : Comparaison des performances des observateurs Kalman et LMI
N.B : ε_{Φ_r} : Erreur d'orientation

4.2.1.3.4. La stabilité :

Nous n’analysons la stabilité d’un filtre de Kalman optimal qu’en régime permanent. En effet, les gains de Kalman dépendent des paramètres extérieurs (ici la vitesse de rotation) mais aussi du temps. Ainsi, l’utilisation des outils d’analyse des systèmes linéaires n’est applicable que lorsque les gains ont atteint leur régime permanent.

La comparaison du lieu des pôles entre un estimateur et le filtre de Kalman optimal montre que :

1. les pôles relatifs au flux rotorique ont très peu bougés comparés aux pôles liés au courant statorique.
2. les pôles liés au courant statorique sont presque situés au centre du cercle unité. Le “système observateur” possède donc une stabilité maximale (à condition que les autres pôles aient aussi un module strictement inférieur à un).

4.3. Algorithme simple et optimal du filtre de Kalman

Les recherches scientifiques sur les observateurs d’état abordent peu les problèmes d’implantation. Les concepteurs des observateurs ne s’occupent pas beaucoup de la complexité algorithmique des commandes et des estimateurs, mais plus à leurs performances. Ensuite, l’algorithme est implémenté sur un DSP à virgule flottante peut alors perdre ses performances et les concepteurs sont alors obligés d’avoir recours à d’autres algorithmes équivalents mais de meilleures performances (*robustesse numérique, stabilité numérique, complexité algorithmique, ...*).

L’étude des structures [11] des matrices utilisées permet de simplifier l’algorithme de filtrage de Kalman, de réduire le coût algorithmique, et par la même occasion améliorer la stabilité numérique du filtre.

4.3.1. Étape de prédiction

4.3.1.1 Prédiction des états : La prédiction est obtenue à partir des propriétés des matrices de transition et de commandes discrètes :

$$\begin{cases} (\hat{i}_{s\alpha})_{pred} = a_{11}\hat{i}_{s\alpha} + b_{11}\hat{i}_{s\beta} + a_{12}\hat{\phi}_{r\alpha} + b_{12}\hat{\phi}_{r\beta} + a_1V_{s\alpha} \\ (\hat{i}_{s\beta})_{pred} = -b_{11}\hat{i}_{s\alpha} + a_{11}\hat{i}_{s\beta} - b_{12}\hat{\phi}_{r\alpha} + a_{12}\hat{\phi}_{r\beta} + a_1V_{s\beta} \\ (\hat{\phi}_{r\alpha})_{pred} = a_{21}\hat{i}_{s\alpha} + b_{21}\hat{i}_{s\beta} + a_{22}\hat{\phi}_{r\alpha} + b_{22}\hat{\phi}_{r\beta} + a_2V_{s\alpha} \\ (\hat{\phi}_{r\beta})_{pred} = -b_{21}\hat{i}_{s\alpha} + a_{21}\hat{i}_{s\beta} - b_{22}\hat{\phi}_{r\alpha} + a_{22}\hat{\phi}_{r\beta} + a_2V_{s\beta} \end{cases} \quad (4.7)$$

4.3.1.2. Calcul de la variance de l'erreur de prédiction $P[k+1/k]$.

La matrice de variance-covariance d'erreur de prédiction est une matrice qui possède les propriétés suivantes: définie par l'équation (3.53) de dimension 4×4 , symétrique, définie positive. Dans ce cas particulier qui peut être simplifié lorsque trois mesures de courants sont réalisées puisque certains termes deviennent alors nuls et d'autres sont égaux (voir équation 3.54). Ainsi, ces termes ont les propriétés particulières : P_{11} , P_{22} et P_{33} , P_{44} sont égaux (respectivement), signifiant que les axes (α, β) jouent un même rôle. De plus, les termes P_{12} et P_{34} sont nuls, les composantes du flux rotorique $\phi_{r\alpha}$ et $\phi_{r\beta}$, courant statorique $i_{s\alpha}$ et $i_{s\beta}$ respectivement, sont donc statistiquement orthogonales.

Ceci est prévisible puisque la matrice de variance des bruits de mesure R fait apparaître deux zéros sur l'antidiagonale. Cette constatation ne peut être établie que dans le cas où trois mesures de courants sont réalisées puisque R est une matrice diagonale. Ceci permet une simplification algorithmique.

En pratique, deux mesures de courant sont effectuées. Les matrices de variance-covariance d'erreur de prédiction et d'estimation ne comportent pas de propriété structurelle intéressante pour une optimisation du coût de calcul. Ainsi, les calculs du filtre supposent couramment que trois mesures de courant sont effectuées.

$$P[k/k] = \begin{bmatrix} P_{11}[k/k] & P_{12}[k/k] & P_{13}[k/k] & P_{14}[k/k] \\ P_{21}[k/k] & P_{22}[k/k] & P_{23}[k/k] & P_{24}[k/k] \\ P_{31}[k/k] & P_{32}[k/k] & P_{33}[k/k] & P_{34}[k/k] \\ P_{41}[k/k] & P_{42}[k/k] & P_{43}[k/k] & P_{44}[k/k] \end{bmatrix} \quad (4.8)$$

$$P[k/k] = \begin{bmatrix} P_{11}[k/k] & 0 & P_{13}[k/k] & P_{14}[k/k] \\ 0 & P_{11}[k/k] & -P_{14}[k/k] & P_{13}[k/k] \\ P_{13}[k/k] & -P_{14}[k/k] & P_{33}[k/k] & 0 \\ P_{14}[k/k] & P_{13}[k/k] & 0 & P_{33}[k/k] \end{bmatrix} \quad (4.9)$$

Les propriétés structurelles particulières de la matrice telles que les symétries, antisymétries et zéros dans la matrice de variance-covariance d'erreur d'estimation sont conservées dans la matrice de variance-covariance d'erreur de prédiction et permettent par la suite de réduire la complexité algorithmique puisque le nombre de valeurs distinctes est réduit à quatre coefficients :

$$P[k+1/k] = \begin{bmatrix} P_{11}[k+1/k] & 0 & P_{13}[k+1/k] & P_{14}[k+1/k] \\ 0 & P_{11}[k+1/k] & -P_{14}[k+1/k] & P_{13}[k+1/k] \\ P_{13}[k+1/k] & -P_{14}[k+1/k] & P_{33}[k+1/k] & 0 \\ P_{14}[k+1/k] & P_{13}[k+1/k] & 0 & P_{33}[k+1/k] \end{bmatrix} \quad (4.10)$$

$$\left\{ \begin{array}{l} P_{11}[k+1/k] = (a_{11}^2 + b_{11}^2)P_{11}[k/k] + 2(a_{11}a_{12} + b_{11}b_{12})P_{13}[k/k] \\ \quad + 2(a_{11}b_{12} - a_{12}b_{11})P_{14}[k/k] + (a_{11}^2 + b_{11}^2)P_{33}[k/k] + \alpha \\ \\ P_{13}[k+1/k] = (a_{21}a_{11} + b_{21}b_{11})P_{11}[k/k] \\ \quad + (a_{12}a_{21} + b_{12}b_{21} + a_{11}a_{22} + b_{11}b_{22})P_{13}[k/k] \\ \quad + (a_{21}b_{12} - a_{12}b_{21} + a_{11}b_{22} - a_{22}b_{11})P_{14}[k/k] \\ \quad + (a_{12}a_{22} + b_{12}b_{22})P_{33}[k/k] \\ \\ P_{14}[k+1/k] = (a_{21}a_{11} + b_{21}b_{11})P_{11}[k/k] \\ \quad + (a_{12}a_{21} + b_{12}b_{21} + a_{11}a_{22} + b_{11}b_{22})P_{13}[k/k] \\ \quad + (a_{21}b_{12} - a_{12}b_{21} + a_{11}b_{22} - a_{22}b_{11})P_{14}[k/k] \\ \quad + (a_{12}a_{22} + b_{12}b_{22})P_{33}[k/k] \\ \\ P_{33}[k+1/k] = (a_{11}^2 + b_{11}^2)P_{11}[k/k] + 2(a_{21}a_{22} + b_{21}b_{22})P_{13}[k/k] \\ \quad + 2(a_{21}b_{22} - a_{22}b_{21})P_{14}[k/k] + (a_{22}^2 + b_{22}^2)P_{33}[k/k] + \beta \end{array} \right. \quad (4.11)$$

Remarque: la structure matricielle à l'instant k défini par l'équation (3.53) est conservée à l'instant $k+1$ comme il a été démontré. Ainsi, quelle que soit la valeur de k , les matrices sont définies chacune par quatre termes uniquement au lieu de seize.

4.3.2. Étape de correction

4.3.2.1. Calcul des gains de correction $k[k+1]$.

La matrice de gain $k[k+1]$ est définie par une matrice de dimension 4×2 , et présente des coefficients communs permettant la réduction des calculs par :

$$k[k+1] = \begin{bmatrix} k_{11}[k+1] & -k_{12}[k+1] \\ k_{12}[k+1] & k_{11}[k+1] \\ k_{13}[k+1] & -k_{14}[k+1] \\ k_{14}[k+1] & k_{13}[k+1] \end{bmatrix} \quad (4.12)$$

$$\text{avec : } \begin{cases} k_{11}[k+1] = P_{11}[k+1/k]/(P_{11}[k+1/k]+1) \\ k_{12}[k+1] = 0 \\ k_{13}[k+1] = P_{13}[k+1/k]/(P_{11}[k+1/k]+1) \\ k_{14}[k+1] = P_{14}[k+1/k]/(P_{11}[k+1/k]+1) \end{cases} \quad (4.13)$$

Le coût de calcul algorithmique de la matrice de correction est devenu très faible par rapport à celui calculé par l'expression brute du filtre de Kalman classique vue auparavant qu'est $[k+1] = PC'(CPC'+R)^{-1}$, où il est nécessaire de réaliser plusieurs multiplications et inversions de matrices. De plus le gain k_{12} est nul; il n'y a donc pas lieu de le calculer à chaque période d'échantillonnage. Ceci simplifie notre algorithme.

4.3.2.2. Étape d'estimation des états

L'étape d'estimation des états consiste à comparer les prédictions aux mesures et d'effectuer ensuite une correction adéquate :

$$\left\{ \begin{array}{l} \varepsilon_a = (i_{s\alpha})_{mes} - (\hat{i}_{s\alpha})_{pred} \\ \varepsilon_\beta = (i_{s\beta})_{mes} - (\hat{i}_{s\beta})_{pred} \\ \hat{i}_{s\alpha} = (\hat{i}_{s\alpha})_{pred} + k_{11} \varepsilon_a \\ \hat{i}_{s\beta} = (\hat{i}_{s\beta})_{pred} + k_{11} \varepsilon_\beta \\ \hat{\phi}_{r\alpha} = (\hat{i}_{r\alpha})_{pred} + k_{13} \varepsilon_a + k_{14} \varepsilon_\beta \\ \hat{\phi}_{r\beta} = (\hat{i}_{r\beta})_{pred} - k_{14} \varepsilon_a + k_{13} \varepsilon_\beta \end{array} \right. \quad (4.14)$$

4.3.2.3. Calcul de la variance de l'erreur d'estimation $P[k+1/k+1]$

L'expression particulière (3.59) de la matrice $P[k+1/k+1]$ et la matrice de gain contiennent trois coefficients en commun ce qui permet de réduire le nombre de calculs algorithmiques et le nombre de cases mémoires nécessaires à la réalisation du filtre .

$$P[k+1/k+1] = \begin{bmatrix} P_{11}[k+1/k+1] & 0 & P_{13}[k+1/k+1] & P_{14}[k+1/k+1] \\ 0 & P_{11}[k+1/k+1] & -P_{14}[k+1/k+1] & P_{13}[k+1/k+1] \\ P_{13}[k+1/k+1] & -P_{14}[k+1/k+1] & P_{33}[k+1/k+1] & 0 \\ P_{14}[k+1/k+1] & P_{13}[k+1/k+1] & 0 & P_{33}[k+1/k+1] \end{bmatrix} \quad (4.15)$$

$$\text{avec : } \left\{ \begin{array}{l} P_{11}[k+1/k+1] = k_{11}[k+1] \\ P_{13}[k+1/k+1] = k_{13}[k+1] \\ P_{14}[k+1/k+1] = k_{14}[k+1] \\ P_{33}[k+1/k+1] = P_{33}[k+1/k] (P_{13}[k+1/k] k_{13}[k+1] + P_{14}[k+1/k] k_{14}[k+1]) \end{array} \right. \quad (4.16)$$

Remarque: Le calcul de $P[k+1/k+1]$ prouve que la structure de la matrice $P[k/k]$ est conservée. L'hypothèse de départ relative à la structure de $P[k/k]$ est donc vérifiée par récurrence.

4.3.2.4. Coût algorithmique du filtrage de Kalman optimal

En tenant compte des propriétés particulières de la matrice et des énormes réductions, le nombre d'opérations est simplifié au maximum par la recherche des doublons (notés tmp_i) dans les équations (3.55). Ceci permet une optimisation de notre filtre de Kalman proposé qui est comparé à un filtre de Kalman classique dont l'implémentation utilise les équations brutes, c'est à dire sans prise en compte d'aucune propriété structurelle des matrices. Les résultats sont donnés dans le tableau(3.4).

Le nombre d'opérations pour le filtre de Kalman proposé est réduit à 146 au lieu de 677 opérations nécessaires pour un filtre de Kalman classique programmé de manière "brute". En effet, puisque la matrice de variance-covariance d'erreur de prédiction est une matrice symétrique ($P[k+1/k] = P[k+1/k]'$), les équations matricielles du filtre peuvent s'écrire sous la forme :

$$\begin{cases} k[k+1] = (CP[k+1/k])' (CP[k+1/k]C' + R)^{-1} \\ P[k+1/k+1] = P[k+1/k] - k[k+1](CP[k+1/k]) \end{cases} \quad (4.17)$$

Tableau 4.1: Nombre d'opérations pour le calcul de filtre optimal (valeurs à gauche) et brut

Matrice	Nombre d'opérations de multiplication et d'inversion		Nombre d'addition et de soustraction	
A_d	7		2	
B_d	0		0	
$X[k+1/k]$	20	(24)	16	(20)
$p[k+1/k]$	44	(128)	34	(112)
$k[k+1]$	4	(103)	1	(73)
$p[k+1/k+1]$	2	(96)	2	(80)
$X[k+1/k+1]$	6	(16)	8	(16)
Total	83	(374)	63	(303)

Un programme simple et efficace permet la réduction du coût de calcul. Il va permettre l'implantation du filtre sur des DSP ou microcontrôleurs de plus faibles performances. De plus, il permet d'utiliser le temps restant pour estimer d'autres grandeurs comme les paramètres de la machine.

L'implantation d'un filtre optimisé sur un calculateur numérique est moins complexe qu'une programmation "brute", mais toutefois bien plus complexe dépendant de la manière proposée.

Conclusion

Ce chapitre était l'objet d'une étude structurée et comparative qui a commencé par la modélisation du système avec le choix du référentiel et du repère approprié et se termine par un pré-réglage automatique du filtre de Kalman.

1. Dans la première partie de ce chapitre: nous avons présenté le modèle de la machine asynchrone utilisée d'une façon standard dans le contrôle des machines à courant alternatif. Le modèle qu'on a obtenu est non linéaire. Pour commander de tels systèmes, il est possible de développer des commandes linéaires à partir d'une approximation linéaire de leur modèle au détriment des performances obtenues. De ce fait, il est possible d'utiliser des commandes non linéaires pour obtenir de meilleures performances tant statiques que dynamiques.
2. Dans la deuxième partie de ce chapitre, nous avons présenté les équations d'état qui étaient moins optimisées pour un processeur. Puis, nous avons effectué un développement mathématique pour les adapter et pour qu'ils soient utilisables.
3. L'étude du modèle du système avec le filtrage de Kalman classique d'ordre 1 permet d'aboutir aux bons résultats de simulation mais, nécessite un DSP performant à l'étape de l'implantation. En effet, si nous voulons garder la taille du problème raisonnable et encore obtenir plus de performances avec le minimum d'instruction (DSP raisonnable), nous effectuons le même travail mais avec un filtre de Kalman optimisé d'ordre 2 et un logiciel plus performant dans notre cas Visual C++ .

Cette étude est le résultat d'un développement théorique fin de l'ensemble des structures des matrices qui rend possible la réduction de la complexité algorithmique du filtre de Kalman. Cette réduction [11] est le résultat d'une manipulation théorique des produits matriciels en équations récurrentes et en recherchant dans ces équations de calcul la performance souhaitée.

Conclusion générale :

Le premier chapitre de ce mémoire a consisté à faire l'inventaire d'un certain nombre de problématiques du Génie Électrique pouvant faire appel à l'automatique et à montrer qu'une grande partie de ces problèmes pouvait être abordée au moyen d'outils issus de la commande avancée. Le PID, omniprésent dans les organes de commande des dispositifs électriques pouvait répondre à toutes ces problématiques de manière satisfaisante. Cependant, nous avons mis en évidence que la commande par optimisation du filtre de Kalman généralisé, optimal ouvrait des perspectives très intéressantes pour résoudre certains problèmes clés. Nous nous sommes intéressés en particulier aux problèmes de synthèse d'un observateur de flux pour la machine asynchrone .

Nous avons introduit dans un second temps le formalisme et les résultats essentiels relatifs à la commande optimale par filtrage de Kalman.

Quelques rappels relatifs à la représentation des systèmes et aux notions d'observation associées ont ainsi été développées pour être mis à profit par la suite après avoir formalisé les problématiques de synthèse de l'observateur de flux de Kalman. Nous avons commencé par aborder dans le détail des problèmes de synthèse d'un observateur de flux rotorique pour une machine asynchrone. Une étude bibliographique détaillée a permis de préciser l'intérêt de l'approche préconisée. Le modèle de conception retenu est celui du sous-système électrique représenté dans un repère diphasé fixe obtenu par les transformations de Park-Concordia . C'est en effet un modèle linéaire paramétré par les vitesses mécaniques du rotor à partir duquel nous avons pu définir le problème du filtre de Kalman dont la résolution conduit à l'obtention d'un observateur, lui aussi paramétré par la vitesse mécanique du rotor.

Une analyse détaillée des qualités d'un observateur de flux ainsi conçu a pu être menée à la fois en simulation et implantation. Les performances obtenues sont intéressantes pour un degré de complexité de l'observateur tout à fait acceptable. La discréétisation de l'observateur, un système dynamique, a été réalisée en tenant compte du caractère des différents signaux électriques.

Malheureusement, l'implantation d'un filtre de Kalman traditionnel est difficile sur un processeur classique à cause du nombre important d'opérations nécessaires à chaque période d'échantillonnage pour réactualiser les estimateurs. Nous avons donc étudié la réduction du volume de l'algorithme pour envisager l'implantation du filtre sur des processeurs bon marché.

Enfin, nous soulignons les bonnes propriétés de robustesse de l'observateur vis-à-vis l'erreur de modèle de la machine considérée.

Bibliographie

- [1] **M. Hilairet, F. Auger, C. Darengosse**, "Two efficient Kalman filters for flux and velocity estimation of induction motors," Proc IEEE PESC'00, Vol 2, pp 891-896, juin 2000.
- [2] **C. Darengosse, P. Chevrel and R. Le Doeuff**, "A linear parameter varying flux observer : Design and experimentation", The 31st IEEE Power Electronics Specialists Conference, Galway (Irlande) 2000, CD-ROM Proceedings.
- [3] **G.C.Verghese and S.R.Sanders**, "Observers for flux estimation in induction machines". IEEE Transaction on automatic control. February 1988, vol. 35, n°1, pp85-94.
- [4] **G. Bornard and H.Hammouri**, "A high gain observer for a class of uniformly observable systems", The 30th IEEE Conference on Decision and Control, Brighton (Angleterre) 1991, pp1494-1500.
- [5] **S.Sangwongwanich, T. Yonemoto, T. Furuhashi, and S. Okuma**, "Design of sliding observer for estimation of rotor flux induction motors". Electrical Engineering in Japan, 1988, vol. 110, n°6, pp279-288.
- [6] **C.M. Lee and C.M. Chen**, "Speed Sensorless vector control of induction motor using Kalman-filter-assisted adaptive observer". IEEE transaction on industrial electronics, April 1998, vol. 45, n°2. pp359-362.
- [7] **V.I. Utkin**, "Sliding modes in control and optimization", Communications and control engineering series. Springer Verlag 1992.
- [8] **C. Darengosse and P.Chevrel**, "Synthèse et expérimentation d'un observateur de flux LPV", Les 1ères Journées Doctorales en Automatique 1999, Nancy (France), pp117-120.
- [9] **M.S. Grewal, A.P. Andrews**, "Kalman filtering, theory and practice," *Prentice Hall*, Englewood Cliffs, New Jersey, 1993. GRE93
- [10] **P. Borne, G. Dauphin-Tanguy, J.P. Richard, F. Rotella et I. Zambettakis**, "Commande et optimisation des processus," Méthodes et Techniques de l'Ingénieur, Editions Technip, 1990.

- [11] **M.S. Grewal, A.P. Andrews**, "Kalman filtering, theory and practice," Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [12] **F. Auger**, "Introduction a la théorie du signal et de l'information - cours et exercices" Editions Technip, 1999.
- [13] **S. Bittanti, M. Savaresi**, "On the parametrization and design of an extended Kalman filter frequency tracker," IEEE Transactions on Automatic Control, Vol 45, No 9, pp 1718-1724 , septembre 2000.
- [14] **J.P. Caron et J.P. Hautier**, "Modélisation et commande de la machine asynchrone," Edition Technip-Méthodes et Pratiques de l'ingénieur, 1995 CAR95
- [15] **G. Segier, F. Notelet**, "Electrotechnique Industrielle", Edition Tec&Doc ,
- [16] **J. Lesenne, F. Notelet , G. Segier, F. Notelet**, "Introduction a l'electrotechnique approfondie", Edition Tec&Doc1981
- [17] **P. Vas**, "Sensorless vector and direct torque control," *Oxford University Press*, 1998.
- [18] **W.S, Levine**(éditeur) "The control handbook" IEEE press ,1996.
- [19] **C. Canudas de Wit**, Commande des moteurs asynchrones : Optimisation, discréétisation et observateurs. Systèmes Automatisé : Information-Commande-Communication Editions Hermès 2000.
- [20] **K.J. Åmström and B. Wittenmark**, Computer controlled systems – Theory and design. Prentice Hall International Edition, 1990.
- [21] **P. Apkarian**, "On the discretization of LMI-synthesized parameter-varying controllers". Automatica, 1997, vol. 33, n°4. pp655-661.
- [22] **E. Larroche, E. Sedda, C. Durieu, JP. Louis**, "Erreurs de modélisation d'une Machine asynchrone " Revue internationale de Génie Électrique Vol 3N1
- [23] **C.Manes, F.Prasiliti, M.Tyursini**, " A comparative Study of Rotor Flux Estimation in induction motors With a Nonlinear Observer and the extended Kalman filters," Proc IEEE 1994.
- [24] **Kemin Zhou** "Essenciel of Robuste Control "
- [25] LMI control toolbox de matlab
- [26] **G. Sitton**, "A collection of functions for the TMS320C30". Texas Instruments application report SPRA 117, 1997.

[27] **A. Dell Aquila, F. Cupertino, L. Salvatore, S. Stasi** " Kalman filters Estimators Applier to Robust Control of induction motors Drives" IEEE 1998.

[28] **A.Dell Aquila ,F.Cupertino,L.Salvatore,S.Stasi** " Kalman filters Estimators Applier to Robust Control of induction motors Drives" IEEE 1998.

[29] **K.D. Hurst et al**, "A self tuning closed-loop flux observer for sensorless control of induction machines," IEEE Transactions on Power Electronics, Vol 12, No 5, pp 807-815, sept 1997.

[30] **K. Jamsas, L. Klander**, "la bible du programmeur en C/C," Editions Eyrolles, 1999.

[31] **P.L. Jansen, R.D. Lorenz**, "Transducerless position and velocity estimation in induction AC ma-chines,"IEEE Transactions on Industry Applications, Vol 31, No 2, pp 240-247, mars 1995. JAN95

[32] **P. L. Jansen, M. J. Corley, R. D. Lorenz**, "Flux, position, and velocity estimation in AC machines at zero and low speed via tracking of high frequency saliencies," *Jounal EPE*, Vol 9, No 1-2, pp 45-50,Novembre 1999. JAN99

[33] **H. Kubota, K. Matsuse, T. Nakano**, "DSP-based speed adaptive flux observer of induction mo-tor,"IEEE Transactions on Industry Applications, Vol 29, No 2, pp 152-156, Mars/Avril 1993.

[34] **H. Kubota, K. Matsuse**, "Speed sensorless field-oriented control of induction motor with rotor resistance adaptation," IEEE Transactions on Industry Applications, Vol 30, No 5, pp 1219-1224,Septembre/Octobre 1994.

[35] **J. Maes et J. Melkerbeek**, "Speed-sensorless direct torque control of induction motors using an adaptive flux observer," IEEE Transactions on Industry Applications, Vol. 36, No. 3, pp 778-785,Mai-Juin 2000.

[36] **L.P. Maguire, G.W. Irwin**, "Transputer implementation of Kalman filters," *IEE Proceedings D*,Vol. 138, No. 4, pp 355-362, Juillet 1991. MAG91

[37] **R. Marino, S. Peresada, P. Tomei**, "On-line stator and rotor resistance estimation for induction motors," IEEE Transactions on Control Systems Technology, Vol. 8, No. 3, pp 570-579, Mai 2000. MAR00

[38] **J. Reekie**, "Realtime DSP: the TMS320C30 course," revision 3, <http://www.ti.com>, Fevrier 1994.

[39] **K. Reif, R. Unbehauen**, "The extended Kalman filter as an exponential observer for nonlinear systems," IEEE Transactions on Signal Processing, Vol. AC-47, No. 8, pp 2324-2328, August 1999.

[40] **X. Roboam, B. De Fornel, J.M. Semenzato**, "A comparative analysis of stochastic and deterministic techniques applied to an induction motor drive," *Proc ICEM*, 1992.

[41] **H. Tajima, Y. Hori**, "Speed sensorless field-orientation control of the induction machine," IEEE Transactions on Industry Applications, Vol. 12, No. 4, pp 175-180, Janvier-Fevrier 1993.

[42] **A. Tanaka**, "Parallel computation in linear discrete filtering," IEEE Transactions on automatic control, Technical notes and correspondance, vol. AC-20, p 573-575, 1975.

[43] **L.C. Zai, C.L. DeMarco, T.A. Lipo**, "An extended Kalman filter approach to rotor time constant measurement in PWM induction motor drives," IEEE Transactions on Industry Applications, Vol. 28, No. 1, pp 96-104, Janvier-Fèvrier 1992.204.

[44] **I. Zein, L. Loron**, "A second-order model for the real-time identification of induction motor," *Electrimacs '99*, Vol. 3, pp 99-104, Lisbonne, Septembre 1999.

[45] **S. Boyd, L. El Gahoui, E. Feron and V. Balakrishnan** Linear matrix inequalities in system and control theory.. SIAM studies in applied mathematics 1994.

[46] "H . design with pole placement constraints: an LMI approach". **M. Chilali and P. Gahinet**. IEEE transactions on automatic control, March 1996, vol. 41, n°3, pp358-367.

[47] **M. Dettori and C.W. Scherer** "Design and implementation of a gain-scheduled controller for a compact disc player". The European Control Conference 1999. Karlsruhe (Germany). CDROM Proceedings.

[48] **J.C. Doyle, K. Glover, Khargonekar and B.A. Francis** "State-space solution to standard H_2 and H_∞ control problems".. IEEE transactions on automatic control,(1989), vol. 34, n°8, pp831-846.

[49] **P.Gahinet, P. Apkarian and M. Chilali**. "Affine parameter-dependant Lyapunov functions and real parametric uncertainty". IEEE transactions on automatic control, March 1996, vol. 41, n°3, pp436-442.

[50] **P. Gahinet, A. Nemirovsky, A.J. Laub** and **M. Chilali**, LMI Control Toolbox. Mai1995, Mathworks Inc.

[51] **C. Scherer** The Riccati Inequality and State-Space H^∞ . Optimal Control. Ph. D. Thesis1990. Würzburg University.

[52] **H.K. Khalil** and **E.G. Strangas** "Robust speed control of induction motors using position an current measurements". IEEE transaction on automatic control. August 1996, vol.41, n°8. pp1216-1220.

[53] **Anders Helmersson**, Methods for Robust Gain Scheduling. Linköping Studies in Science and Technology, dissertation n° 406, 1995.

[54] **A. Djermoune** and **P. Goureau** "State feedback linearizing of non-linear induction motor control". Electrimacs'96, Saint-Nazaire (France), September 1996, p p. 385-390.

[55] **T. Du, P. Vas** "Application of Extended observers for joint state and parameter estimation in high-performance AC drives" IEE Proc Appl .Vol .142, Nop.2 pp 71-77, Mai 1995.

Annexes

Annexe1

- Programme en visual C++ sur le filtre de kalman d'ordre 2

```
//-----  
// Program: kalman.cpp  
//-----  
#include <iostream.h>  
#include <stdio.h>  
#include <iomanip.h>  
#include <stdlib.h>  
#include <time.h>  
#include <math.h>  
#include <fstream.h>  
#include <iomanip.h>  
  
#define Vmax 281.7 // Tension max aux bornes du moteur Vmax=2*220*sqrt(2/3)  
#define sqrt2sur3 0.81649658092773 // sqrt(2/3)  
#define _1sursqrt6 0.40824829046386  
#define _1sursqrt2 0.70710678118655 // 1/sqrt(2)  
  
#define Imax 5.5726  
#define sqrt3sur2 1.22474487139159 // sqrt(3/2)  
#define sqrt2 1.41421356237310 // sqrt(2)  
#define a11 0.8935009848  
#define b11_1 0.0000082667  
#define a12_1 0.2209176997  
#define a12_2 0.0000033333  
#define b12_1 0.0156310606  
  
#define a21 0.0009331564  
#define b21_1 -0.0000001984  
#define a22_1 0.9944960055  
#define a22_2 -0.0000000800  
#define b22_1 -0.0003894788  
  
#define a1 0.0157250000  
#define a2 0.0000082667
```

```

#define _2pi      6.28318530717959      // 2*pi
#define qsur2     0.00306796157577      // q/2 avec q=2*pi/2^N et N=11 bit
#define sqrt2sur3 0.81649658092773      /* sqrt(2/3) */
#define sqrt3sur2 0.86602540378444      /* sqrt(3)/2 */

//-----
// Function prototype
void machine_parameters(double *);
void euler(double *, double, double, double, double, double, int, double, double,
double, double, double);
void runge_kutta(double *, double, double, double, double, double, int, double,
double, double, double);
void commande(double&, double&, double, double, double *, double, double);
void decalage(void);
void courant(double&, double&, double, double);
void kalman(double *, double *, double&, double *, int, double, double, double,
double);
void vitesse(double *, double);
//-----
// Function prototype
void simul(char *, double, double, int, double, double);
//-----
// Function prototype
void modele(double *, double *, double, double, double, double, double, int, double,
double, double, double, double);
// Function prototype
void modele(double *, double *, double, double, double, double, double, int, double,
double, double, double, double);

//-----
// ADRESSES ET DONNEES RELATIF A LA COMMANDE VECTORIELLE
//-----
int pp=2;                                // Nombre de paire de pole
double Te=400E-6;                          // Periode d'echantillonnage
double Fe=2500.0;                          // Frequence d'echantillonnage
double pi=3.14159265358979;
double Omegas[2]={0.0, 0.0};
double Thetas;
double costhetas, sinthetas;
double Vsdref[3]={0.0, 0.0, 0.0};
double Vsdrefp[3]={0.0, 0.0, 0.0};

```

```

double Esq[3]={0.0, 0.0, 0.0};
double Isd[2]={0.0, 0.0};
double Isdref[3]={0.0, 0.0, 0.0};           // Consigne du courant selon l'axe d
double Vsqref[3]={0.0, 0.0, 0.0};
double Vsqrefp[3]={0.0, 0.0, 0.0};
double Esd[3]={0.0, 0.0, 0.0};
double Isq[2]={0.0, 0.0};
double Isqref[3]={0.0, 0.0, 0.0};           // Consigne du courant selon l'axe q

double Vs1ref, Vs2ref;
double Vsalpha, Vsbeta;
double Frobs[2]={0.0, 0.0};
double Isdrefp[3]={0.0, 0.0, 0.0};
double Frref[3]={0.0, 0.0, 0.0};

double Isqrefp[3]={0.0, 0.0, 0.0};
double vitref[3]={0.0, 0.0, 0.0};

double Coeff_id[12]={1.2375667836, -0.3828928860, 13.4288728723, -16.6191270082,
5.1418198895, -13.5703068633, 11.6187411096, -1.0000000000, 1.3316527913, -
0.3316527913, 0.0940860077, 0.0512400947};
double Coeff_iq[12]={1.7042875779, -0.7261490371, 4.8387798683, -8.2466724220,
3.5136753420, -1.1872435757, 1.0814607873, -1.0000000000, 1.7381068611, -
0.7381068611, 0.0338192831, -0.0119578240};
double Coeff_f[9]={1.9062675742, -0.9084640161, 40.1253306175, -76.4896166584,
36.4524189989, -6.7638424070, 6.6757094490, -0.0163588366, 0.0185552786};
double Coeff_v[9]={1.9682546401, -0.9685065821, 0.0896462702, -0.1764466874,
0.0868230028, -0.0056056537, 0.0055830680, -0.0073513229, 0.0076032648};

//-----
// DIVERS
//-----

double tmp;                                // variable temporaire (relle)
volatile unsigned int toto;                  // variable temporaire (entier)
int temps;        // temps = indication du temps pour executer l'algorithme
double Isalpha, Isbeta;
double Is1, Is2;

```

```

//-----
// Program: param_kf.h
//-----
double      b11;
double      a12;
double      b12;
double      b21;
double      a22;
double      b22;

double  vit2, vite;

double Xobs[4] ={0.0, 0.0, 0.0, 0.0};
double Xpred[4]={0.0, 0.0, 0.0, 0.0};

double Ppred[4], Pest[4]={0.0, 0.0, 0.0, 0.0};
double Innovation[2];
double tmp1, tmp2, tmp3, tmp4;
double alpha1=400.0, alpha2=6.0E-2;
double Cemobs;

//-----
// Program: param_mas.h
//-----

int  n=0;
double t=0.0;

double a_mot, b_mot, c_mot, km_mot, m1, m2;
double rs_mot, rr_mot, ls_mot, lr_mot, msr_mot, jm_mot, fm_mot;
double sg_mot, Tr_mot;

double ThetaQ, ThetaSat, vitest;
double X_mot[6]={0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
double Cr=0.0;
double Cem_mot, Fr_mot;
double Isa_mot, Isb_mot;
double Vsa, Vsb;

double rs, rr, ls, lr, msr, jm, fm;

```

```

// variable externe
//double Te;           // Période d'échantillonnage
//int pp;              // Nombre de paire de pole
double Fr;

// Référence (alias)
double& alias_Vsa=Vsa;
double& alias_Vsb=Vsb;

double& alias_Isalpha=Isalpha;
double& alias_Isbeta=Isbeta;

double& alias_Cemobs=Cemobs;

double vit[2]={0.0, 0.0};

//-----
// Program: param_vitesse.h
//-----

float X[3]={0, 0, 0};
float kimacs[3]={0.2218, 0.0278, 0.0017};

//double Fe;           // Fréquence d'échantillonnage

//-----
void kalman(double Xobs[], double Frobs[], double& Cemobs, double vit[], int pp,
double Vsalpha, double Vsbeta, double Isalpha, double Isbeta)
{
debut:
// Calcul des coefficients de Ad */
vite=pp*vit[1];           /* Vitesse du rotor en grandeur electrique */
vit2=vite*vite;           /* vite2=omega_m^2=(p*Omega)^2 */

b11=b11_1*vite;
a12=a12_1+a12_2*vit2;
b12=b12_1*vite;

b21=b21_1*vite;
a22=a22_1+a22_2*vit2;
b22=b22_1*vite;

```

```

// Prediction */
Xpred[0]= a11*Xobs[0]+b11*Xobs[1]+a12*Xobs[2]+b12*Xobs[3]+a1*Vsalpha;
Xpred[1]=-b11*Xobs[0]+a11*Xobs[1]-b12*Xobs[2]+a12*Xobs[3]+a1*Vsbeta;
Xpred[2]= a21*Xobs[0]+b21*Xobs[1]+a22*Xobs[2]+b22*Xobs[3]+a2*Vsalpha;
Xpred[3]=-b21*Xobs[0]+a21*Xobs[1]-b22*Xobs[2]+a22*Xobs[3]+a2*Vsbeta;

Ppred[0]=(a11*a11+b11*b11)*Pest[0]+2*(a11*a12+b11*b12)*Pest[1]+2*(a11*b12-
a12*b11)*Pest[2]+(a12*a12+b12*b12)*Pest[3]+alpha1;

tmp1=a12*a21+b12*b21;
tmp2=a11*a22+b22*b11;
tmp3=a21*b12-a12*b21;
tmp4=a11*b22-a22*b11;

Ppred[1]=(a21*a11+b21*b11)*Pest[0]+(tmp1+tmp2)*Pest[1]+(tmp3+tmp4)*Pest[2]+(a1
2*a22+b12*b22)*Pest[3];
Ppred[2]=(a21*b11-a11*b21)*Pest[0]+(tmp3-tmp4)*Pest[1]+(tmp2-
tmp1)*Pest[2]+(a22*b12-a12*b22)*Pest[3];
Ppred[3]=(a21*a21+b21*b21)*Pest[0]+2*(a21*a22+b22*b21)*Pest[1]+2*(a21*b22-
a22*b21)*Pest[2]+(a22*a22+b22*b22)*Pest[3]+alpha2;

// Correction */
tmp1=1/(Ppred[0]+1);
Pest[0]=Ppred[0]*tmp1; // K11
Pest[1]=Ppred[1]*tmp1; // K13
Pest[2]=Ppred[2]*tmp1; // K14
Pest[3]=Ppred[3]-(Ppred[1]*Pest[1]+Ppred[2]*Pest[2]);

Innovation[0]=Isalpha-Xpred[0];
Innovation[1]=Isbeta -Xpred[1];

Xobs[0]=Xpred[0]+Pest[0]*Innovation[0];
Xobs[1]=Xpred[1]+Pest[0]*Innovation[1];
Xobs[2]=Xpred[2]+Pest[1]*Innovation[0]-Pest[2]*Innovation[1];
Xobs[3]=Xpred[3]+Pest[2]*Innovation[0]+Pest[1]*Innovation[1];

Cemobs=pp*(Xobs[1]*Xobs[2]-Xobs[0]*Xobs[3]);
Frobs[0]=sqrt(Xobs[2]*Xobs[2]+Xobs[3]*Xobs[3]);
}

```

```

//-----
// Program: commande.cpp
//-----

//-----
void commande(double& Vsalpha, double& Vsbeta, double Isalpha, double Isbeta,
double vit[],
    double Fr, double t)
{
    Frobs[0]=Fr;

    // COMMANDE VECTORIELLE DIRECTE
    //-----
    // TRANSFORMATION DE PARK
    Omegas[0]=pp*vit[0]+2.48*Isq[0]/(Frobs[0]+1e-6);           // Msr/Tr=2.48
    Thetas=Thetas+0.5*Te*(Omegas[0]+Omegas[1]);
    Thetas=Thetas-floor(Thetas/(2*pi))*2*pi;

    costhetas=cos(Thetas);
    sinthetas=sin(Thetas);

    // PARK
    Isd[0]= costhetas*Isalpha+sinthetas*Isbeta;
    Isq[0]=-sinthetas*Isalpha+costhetas*Isbeta;

    //-----
    // LOI DE REGULATION DU FLUX ROTORIQUE
    Frref[0]=0.6;

    Isdrefp[0]=Coeff_f[0]*Isdrefp[1]+Coeff_f[1]*Isdrefp[2]
    +Coeff_f[2]*Frref[0]+Coeff_f[3]*Frref[1]+Coeff_f[4]*Frref[2]
    +Coeff_f[5]*Frobs[0]+Coeff_f[6]*Frobs[1]
    +Coeff_f[7]*Isdref[1]+Coeff_f[8]*Isdref[2];

    // Saturation de Isd
    if (fabs(Isdrefp[0])<=Imax)
        Isdref[0]=Isdrefp[0];
    else if (Isdrefp[0]>Imax)
        Isdref[0]= Imax;
    else
        Isdref[0]=-Imax;
}

```

```

//-----
// LOI DE REGULATION DE LA VITESSE
if (t<0.2)
  vitref[0]=0.0;
else if (t<3.0)
  vitref[0]=500.0;
else
  vitref[0]=-500.0;

vitref[0]=vitref[0]*2*pi/60;

Isqrefp[0]=Coeff_v[0]*Isqrefp[1]+Coeff_v[1]*Isqrefp[2]
+Coeff_v[2]*vitref[0]+Coeff_v[3]*vitref[1]+Coeff_v[4]*vitref[2]
+Coeff_v[5]*vitref[0]+Coeff_v[6]*vitref[1]+Coeff_v[7]*Isqrefp[1]+Coeff_v[8]*Isqrefp[2];

// Saturation de Isq
tmp=sqrt(Imax*Imax-Isdref[0]*Isdref[0]);

if (fabs(Isqrefp[0])<=tmp)
  Isqref[0]=Isqrefp[0];
else if (Isqrefp[0]>tmp)
  Isqref[0]= tmp;
else
  Isqref[0]=-tmp;

//-----
// CALCUL DES COMPENSATIONS
// Esd[0]=0.0;
// Esq[0]=0.0;

Esd[0]=-(pp*vitref[0]*Frobs[0]+0.0240*Omegas[0]*Isd[0]);
Esq[0]=14.0909*Frobs[0]+0.0240*Omegas[0]*Isq[0];

//-----
// LOI DE REGULATION DU COURANT SELON L'AXE D
Vsdrefp[0]=Coeff_id[0]*Vsdrefp[1]+Coeff_id[1]*Vsdrefp[2]
+Coeff_id[2]*Isdref[0]+Coeff_id[3]*Isdref[1]+Coeff_id[4]*Isdref[2]
+Coeff_id[5]*Isd[0]+Coeff_id[6]*Isd[1]
+Coeff_id[7]*Esq[0]+Coeff_id[8]*Esq[1]+Coeff_id[9]*Esq[2]
+Coeff_id[10]*Vsdref[1]+Coeff_id[11]*Vsdref[2];

```

```

// Saturation de Vsd
if (fabs(Vsdrefp[0])<=Vmax)
  Vseref[0]=Vsdrefp[0];
else if (Vsdrefp[0]>Vmax)
  Vseref[0]= Vmax;
else
  Vseref[0]=-Vmax;

//-----
// LOI DE REGULATION DU COURANT SELON L'AXE Q
Vsqrefp[0]=Coeff_iq[0]*Vsqrefp[1]+Coeff_iq[1]*Vsqrefp[2]
+Coeff_iq[2]*Isqref[0]+Coeff_iq[3]*Isqref[1]+Coeff_iq[4]*Isqref[2]
+Coeff_iq[5]*Isq[0]+Coeff_iq[6]*Isq[1]
+Coeff_iq[7]*Esd[0]+Coeff_iq[8]*Esd[1]+Coeff_iq[9]*Esd[2]
+Coeff_iq[10]*Vsqref[1]+Coeff_iq[11]*Vsqref[2];

// Saturation de Vsq
tmp=sqrt(Vmax*Vmax-Vseref[0]*Vseref[0]);

if (fabs(Vsqrefp[0])<=tmp)
  Vsqref[0]=Vsqrefp[0];
else if (Vsqrefp[0]>tmp)
  Vsqref[0]=tmp;
else
  Vsqref[0]=-tmp;

//-----
// TRANSFORMATION DE PARK INVERSE
Vsalpha=costhetas*Vseref[0]-sinthetas*Vsqref[0];
Vsbeta =sinthetas*Vseref[0]+costhetas*Vsqref[0];

// Vsalpha=50*cos(2*pi*20*t);
// Vsbeta =50*cos(2*pi*20*t-0.5*pi);

// TRANSFORMATION DE CLARKE INVERSE
Vs1ref=Vsalpha*sqrt2sur3;
Vs2ref=-_1sursqrt6*Vsalpha+_1sursqrt2*Vsbeta;

}

```

```

//-----
void courant(double& Isalpha, double& Isbeta, double Is1, double Is2)
{
    Isalpha=sqrt3sur2*Is1;
    Isbeta=sqrt2*(0.5*Is1+Is2);
}

void decalage(void)
{
    // DECALAGE
    Vseref[2]=Vseref[1];
    Vsqref[2]=Vsqref[1];
    Vseref[1]=Vseref[0];
    Vsqref[1]=Vsqref[0];

    Vseref[2]=Vseref[1];
    Vsqref[2]=Vsqref[1];
    Vseref[1]=Vseref[0];
    Vsqref[1]=Vsqref[0];

    Esd[2]=Esd[1];
    Esq[2]=Esq[1];
    Esd[1]=Esd[0];
    Esq[1]=Esq[0];

    Isd[1]=Isd[0];
    Isq[1]=Isq[0];

    Iseref[2]=Iseref[1];
    Isqref[2]=Isqref[1];

    Iseref[1]=Iseref[0];
    Isqref[1]=Isqref[0];

    Iseref[2]=Iseref[1];
    Iseref[1]=Iseref[0];

    Isqref[2]=Isqref[1];
    Isqref[1]=Isqref[0];
}

```

```

Frref[2]=Frref[1];
Frref[1]=Frref[0];

vitref[2]=vitref[1];
vitref[1]=vitref[0];

Frobs[1]=Frobs[0];

vit[1]=vit[0];

Omegas[1]=Omegas[0];

}

//-----
// Program: euler.cpp
//-----

void euler(double X_mot[], double h, double rs_mot , double rr_mot , double a_mot,
double b_mot,
        double c_mot , int pp , double km_mot , double Cr, double fm_mot , double
jm_mot,
        double Vsalpha, double Vsbeta)
{
    int k;
    double res[6];

    modele(res, X_mot, rs_mot, rr_mot, a_mot, b_mot, c_mot, pp, km_mot, Cr, fm_mot,
jm_mot, Vsalpha, Vsbeta);
    for (k=0;k<6;k++)
    {
        X_mot[k]=X_mot[k]+h*res[k];
    }
}

```

```

// Program: machine_parameters.cpp

//-----

void machine_parameters(double parameters[])
{
    parameters[0]=4.3;          // rs
    parameters[1]=2.48;         // rr
    parameters[2]=0.2;          // ls
    parameters[3]=0.176;        // lr
    parameters[4]=parameters[3]; // msr
    parameters[5]=5.4e-3;       // jm
    parameters[6]=1.6e-3;       // fm
}

//-----

void main(int argc, char *argv[])
{
// clock_t start, stop;
// start=clock(); ;

    if (argc!=7)
    {
        cout << "\nvous devez entrer la syntaxe complete \n";
        cout << "Exemple: kalman G.TXT 2 0.0004 1 1 1.3 [enter]";
        cout << endl;
        cout << "fichier de sauvegarde: G.TXT" << endl;
        cout << "temps de simulation : 2 " << endl;
        cout << "pas d'integration : 0.0004" << endl;
        cout << "methode resolution : 1=Euler, 2=Runge-Kutta" << endl;
        cout << "variation de Rs : 1 (+00%)" << endl;
        cout << "variation de Rr : 1.3 (+30%)" << endl;
        putchar(7); // Haut-parleur
        exit(1);
    }

    // Begin of the program
    cout << "Simulation of the Induction machine, Field Oriented Control and Observer" <<
    endl;
    cout << "khaila, GEi3824, TRIVERES" << endl;
    cout << "Version 1.0, 28/10/2000" << endl;
}

```

```

// temps maximal de la simulation
double tmax;
tmax=atof(argv[2]);
if (tmax<=0.0)
{
    cout << "\nvous devez entrer un temps de simulation positif" << endl;
    putchar(7); // Haut-parleur
    exit(1);
}

// Pas d'integration
double h;
h=atof(argv[3]);
if (h<=0.0)
{
    cout << "\nvous devez entrer un pas de calcul positif" << endl;
    putchar(7); // Haut-parleur
    exit(1);
}
if (tmax<=h)
{
    cout << "\nvous devez entrer un pas de calcul inferieur au temps de simulation" <<
endl;
    putchar(7); // Haut-parleur
    exit(1);
}

// choix de la methode d'integration
int methode;
methode=atoi(argv[4]);
if ((methode!=1)&(methode!=2))
{
    cout << "\nvous devez choisir une methode d'integration valide (1 ou 2)" << endl;
    putchar(7); // Haut-parleur
    exit(1);
}

// variation de rs
double delta_rs_mot;
delta_rs_mot=atof(argv[5]);

```

```

// variation de rr
double delta_rr_mot;
delta_rr_mot=atof(argv[6]);

// Call function simul()
simul(argv[1], tmax, h, methode, delta_rs_mot, delta_rr_mot);

// stop=clock;
// cout << "Temps d'execution du programme: " << (long) (stop-start)/ CLK_TCK << "s"
<< endl;
// End of the program
}

//-----
// Program: modele.cpp
//-----

void modele(double res[], double Xn[], double rs_mot, double rr_mot, double a_mot,
           double b_mot, double c_mot, int pp, double km_mot, double Cr, double fm_mot,
           double jm_mot, double Vsalpha, double Vsbeta)
{
    // bruit_Vsa=10*(random(1000)/1000.0-0.5);
    // bruit_Vsb=10*(random(1000)/1000.0-0.5);

    res[0]=-a_mot*rs_mot*Xn[0]+c_mot*rs_mot*Xn[2]+Vsalpha; // +bruit_Vsa;
    res[1]=-a_mot*rs_mot*Xn[1]+c_mot*rs_mot*Xn[3]+Vsbeta; // +bruit_Vsb;
    res[2]=c_mot*rr_mot*Xn[0]-b_mot*rr_mot*Xn[2]-pp*Xn[4]*Xn[3];
    res[3]=c_mot*rr_mot*Xn[1]-b_mot*rr_mot*Xn[3]+pp*Xn[4]*Xn[2];
    res[4]=(km_mot*(Xn[2]*Xn[1]-Xn[3]*Xn[0])-Cr-fm_mot*Xn[4])/jm_mot;
    res[5]=Xn[4];
}

```

```

//-----
// Program: runge_kutta.cpp
//-----
//-----
void runge_kutta(double X_mot[], double h, double rs_mot, double rr_mot, double
a_mot, double b_mot,
    double c_mot, int pp, double km_mot, double Cr, double fm_mot, double
jm_mot,
    double Vsalpha, double Vsbeta)
{
    int k;
    double Xn[6], k1[6], k2[6], k3[6], k4[6];

    modele(k1, Xn, rs_mot, rr_mot, a_mot, b_mot, c_mot, pp, km_mot, Cr, fm_mot,
jm_mot, Vsalpha, Vsbeta);
    for (k=0;k<6;k++) Xn[k]=X_mot[k]+0.5*h*k1[k];

    modele(k2, Xn, rs_mot, rr_mot, a_mot, b_mot, c_mot, pp, km_mot, Cr, fm_mot,
jm_mot, Vsalpha, Vsbeta);
    for (k=0;k<6;k++) Xn[k]=X_mot[k]+0.5*h*k2[k];

    modele(k3, Xn, rs_mot, rr_mot, a_mot, b_mot, c_mot, pp, km_mot, Cr, fm_mot,
jm_mot, Vsalpha, Vsbeta);
    for (k=0;k<6;k++) Xn[k]=X_mot[k]+h*k3[k];

    modele(k4, Xn, rs_mot, rr_mot, a_mot, b_mot, c_mot, pp, km_mot, Cr, fm_mot,
jm_mot, Vsalpha, Vsbeta);

    for (k=0;k<6;k++)
    {
        X_mot[k]=X_mot[k]+(k1[k]+2*(k2[k]+k3[k])+k4[k])*h/6;
    }
}

//-----
// Program: simul.cpp
//-----
//-----
ofstream GraphFile;

```

```

//-----
// routines sortie GraphFile
void OutGraphInit()
{
    // justification a gauche !
    GraphFile.setf(ios::left, ios::adjustfield);
    GraphFile << setprecision( 6 )
    << "time\t\t\t\t"
    << "Isa\t\t\t\t"
    << "Isb\t\t\t\t"
    << "Fra\t\t\t\t"
    << "Frб\t\t\t\t"
    << "wm\t\t\t\t"
    << "theta\t\t\t\t"
    << "Cem\t\t\t\t"
    << "Cr\t\t\t\t"
    << "Fr\t\t\t\t"
    << "Frest\t\t\t\t"
    << "Cemest\t\t\t\t"
    << "Isaest\t\t\t\t"
    << "Isbest\t\t\t\t"
    << "Fraest\t\t\t\t"
    << "Frbest\t\t\t\t"
    << "Vitest\t\t\t\t"
    << endl;
}

//-----
void OutXGraph()
{
    // temps
    GraphFile << setw(6) << t << "\t\t\t\t";
    // Isa_mot
    GraphFile << setw(6) << Isa_mot << "\t\t\t\t";
    // Isb_mot
    GraphFile << setw(6) << Isb_mot << "\t\t\t\t";
    // Fra
    GraphFile << setw(6) << X_mot[2] << "\t\t\t\t";
    // Frб
}

```

```

GraphFile << setw(6) << X_mot[3] << "\t\t\t\t";
// vitesse
GraphFile << setw(6) << X_mot[4]*60/(2*pi*pp) << "\t\t\t\t";
// position
GraphFile << setw(6) << ThetaQ << "\t\t\t\t";
// Cem
GraphFile << setw(6) << Cem_mot << "\t\t\t\t";
// Cr
GraphFile << setw(6) << Cr << "\t\t\t\t";
// Fr
GraphFile << setw(6) << Fr_mot << "\t\t\t\t";
// Frest
GraphFile << setw(6) << Frobs[0] << "\t\t\t\t";
// Cemest
GraphFile << setw(6) << Cemobs << "\t\t\t\t";
// Isaest
GraphFile << setw(6) << Xobs[0] << "\t\t\t\t";
// Isbest
GraphFile << setw(6) << Xobs[1] << "\t\t\t\t";
// Fraest
GraphFile << setw(6) << Xobs[2] << "\t\t\t\t";
// Frbest
GraphFile << setw(6) << Xobs[3] << "\t\t\t\t";
// Vitest
GraphFile << setw(6) << vitest*60/(2*pi*pp) << endl;
}

//-----
void simul(char OutGraphName[], double tmax, double h, int methode,
          double drs_mot, double drr_mot)
{
    GraphFile.open(OutGraphName, ios::out);
    if (!GraphFile)
    {
        cout << "Error to open file to store data" << OutGraphName << " file \n";
        // exit(1);
    }

    OutGraphInit(); // entete du fichier GraphFile
}

```

```

// Induction machine parameters
double *parameters= new double[7]; // Allocation dynamique de la memoire
if (parameters!=NULL)
    cout << "Attribution de la memoire reussie" << endl;
else
    cout << "new n'a pas pu allouer de la memoire" << endl;

machine_parameters(parameters);
rs=parameters[0];
rr=parameters[1];
ls=parameters[2];
lr=parameters[3];
msr=parameters[4];
jm=parameters[5];
fm=parameters[6];

delete parameters; // deallocation dynamique de la memoire

// Parameters uncertainties
rs_mot=rs*drs_mot;    // rs machine
rr_mot=rr*drr_mot;    // rr machine
ls_mot=ls;
lr_mot=lr;
msr_mot=msr;
jm_mot=jm;
fm_mot=fm;

sg_mot=1-msr_mot*msr_mot/(ls_mot*lr_mot);
a_mot=1/(sg_mot*ls_mot);
b_mot=1/(sg_mot*lr_mot);
c_mot=(1-sg_mot)/(sg_mot*msr_mot);
km_mot=2*msr_mot/(sg_mot*ls_mot*lr_mot);
Tr_mot=lr_mot/rr_mot;
m1=fm_mot/jm_mot;
m2=2/jm_mot;

while(t<=tmax)
{
    if (fabs(t-n*Te)<1e-6)
    {
        n++;
    }
}

```

```

Fr_mot=sqrt(X_mot[2]*X_mot[2]+X_mot[3]*X_mot[3]);
Cem_mot=km_mot*(X_mot[1]*X_mot[2]-X_mot[0]*X_mot[3]);
Isa_mot=a_mot*X_mot[0]-c_mot*X_mot[2];
Isb_mot =a_mot*X_mot[1]-c_mot*X_mot[3];
Is1=sqrt2sur3*Isa_mot;
Is2=sqrt2sur3*(-0.5*Isa_mot+sqrt3sursur2*Isb_mot);

// MODELISATION DU CAPTEUR DE VITESSE
ThetaSat=X_mot[5];
while (ThetaSat<=-qsur2) // SATURATION
{
  ThetaSat=ThetaSat+_2pi;
}

while (ThetaSat>(_2pi-qsur2))
{
  ThetaSat=ThetaSat-_2pi;
}

ThetaQ=ceil(ThetaSat*1024/pi); // TRONCATURE

//-----
// MESURE COURANT
courant(alias_Isalpha, alias_Isbeta, Is1, Is2);

//-----
// MESURE VITESSE
vitesse(vit,ThetaQ*pi/1024);
vitest=vit[0];
// vit[0]=X_mot[4]; // SANS PRISE EN COMPTE DU MODELE DU CAPTEUR
DE VITESSE

//-----
// ESTIMATION DU FLUX ROTORIQUE
kalman(Xobs, Frobs, alias_Cemobs, vit, pp, Vsa, Vsb, Isalpha, Isbeta);

//-----
// COMMANDE
Fr=Fr_mot; // AVEC CAPTEUR DE FLUX
// Fr=Frobs[0]; // SANS CAPTEUR DE FLUX
commande(alias_Vsa, alias_Vsb, Isalpha, Isbeta, vit, Fr, t);

```

```

//-----
// Save data
OutXGraph();

//-----
// DECALAGE
decalage();
}

//-----
if (t<2.0)
  Cr=fm_mot*X_mot[4];
else
  Cr=fm_mot*X_mot[4]+1.0;

// DIFFERENTIAL EQUATION RESOLUTION
if (methode==1)
  euler(X_mot, h, rs_mot, rr_mot, a_mot, b_mot, c_mot, pp, km_mot,
        Cr, fm_mot, jm_mot, Vsa, Vsb); // EULER
else
  runge_kutta(X_mot, h, rs_mot, rr_mot, a_mot, b_mot, c_mot, pp, km_mot,
              Cr, fm_mot, jm_mot, Vsa, Vsb); // FOURTH ORDRE RUNGE-KUTTA
METHODE

t=t+h;
}

// Close file
GraphFile.close();

}

//-----
// Program: vitesse.cpp

//-----
void vitesse(double vit[], double teta)
{
  /* calcul de la vitesse par filtrage de Kalman */
  /* prediction */
  X[0]=X[0]+X[1]+0.5*X[2];
  X[1]=X[1]+X[2];
}

```

```
if ((X[0]-teta)<-4)
  X[0]=X[0]+2*pi;

if ((X[0]-teta)>4)
  X[0]=X[0]-2*pi;

/* correction */
tmp=teta-X[0];
X[0]=X[0]+kimacs[0]*tmp;
X[1]=X[1]+kimacs[1]*tmp;
X[2]=X[2]+kimacs[2]*tmp;
vit[0]=X[1]*Fe;
}
}
```

Annexe2

- Programmes de simulation en matlab/simulink du filtre de Kalman classique stochastique d'ordre 1 sont :

1. Programme en matlab/simulink sur le filtre de Kalman d'ordre 1

```
% kha2.m
```

```
global out G Q R GQG;
global Tr_K Ts_K Lr_K Ls_K M_K Rs_K Rr_K p_K J_K pp;

global x_1 P_1 K P h FI Y ;

%=====khaila=====
P=100*eye(5);
Q=diag([1e-6 1e-6 1e-6 1e-6 1e-2]);
G=diag([1e-6 1e-6 1e-6 1e-6 1e-2]);
R= diag([.001 .001]);
%=====khaila=====
Rs_K = 0.6; % résistance des enroulements statoriques en ohms
Rr_K = 0.6 ; % résistance des enroulements rotoriques
Ls_K = 0.0727; % inductance statorique
Lr_K = 0.0727; % inductance rotorique
M_K = 0.0698; % inductance mutuelle
Ts_K=Ls_K/Rs_K; % constante de temps statorique
J_K = 0.035678 ; % moment d'inertie du rotor kg m2

% bruits sur les mesures
puissance_courant=0.0*1e-7;
puissance_tension=0.0*1e-4;
```

2. %programme1.m _Filtre de kalman implanter dans S-function (fichier.m)_

```
function [sys,x0]=khaila(t,x,u,flag,Ts);

global Tr_K Ts_K Lr_K Ls_K M_K Rs_K Rr_K p_K J_K;
global Q R G x_1 P_1 K P h Y out GQG;

%=====khaila=====
```

```

if flag==0           % initialisation
kha2

K = zeros(5,2);
x0 = zeros(5,1);
sys = [0,5,5,6,0,0];
GQG=G*Q*G;

elseif abs(flag)==2,      % Mise a jours des variables d'état

% calcule. Input et output vecteur du filtre de Kalman
U=[u(1);u(2)];
Y=[u(3);u(4)];

Tr_K=Lr_K/Rr_K;
sg_K=1-M_K^2/(Ls_K*Lr_K);
a_K=1/sg_K/Ls_K;
c_K=(1-sg_K)/sg_K/M_K;
alpha_K=-(a_K*Rs_K+c_K*M_K/Tr_K);
b_K=c_K/Tr_K;
sn_K=M_K/Tr_K;
gama_K=-1/Tr_K;
a11=1+alpha_K*Ts;
a13=b_K*Ts;
a14=c_K*x(5)*Ts;
a31=sn_K*Ts;
a33=1+gama_K*Ts;
a34=-x(5)*Ts;

c1=c_K*x(4)*Ts;
c2=-c_K*x(3)*Ts;
c3=-x(4)*Ts;
c4=x(3)*Ts;
a1=a_K*Ts;

% prédiction      Lh=M

diff_FI=[ a11, 0, a13, a14, c1;
          0, a11, -a14, a13, c2;
          a31, 0, a33, a34, c3;
          0, a31, -a34, a33, c4;
          0,0,0,0,1];

```

```

x_l=[ diff_FI(1,1)*x(1)+ diff_FI(1,2)*x(2)+diff_FI(1,3)*x(3)+ diff_FI(1,4)*x(4);
      diff_FI(2,1)*x(1)+diff_FI(2,2)*x(2)+ diff_FI(2,3)*x(3)+ diff_FI(2,4)*x(4);
      diff_FI(3,1)*x(1)+diff_FI(3,2)*x(2)+ diff_FI(3,3)*x(3)+ diff_FI(3,4)*x(4);
      diff_FI(4,1)*x(1)+ diff_FI(4,2)*x(2)+ diff_FI(4,3)*x(3)+ diff_FI(4,4)*x(4);
      diff_FI(5,5)*x(5)]...
+ [a1*u(1);a1*u(2);0;0;0];

GQG=GQG-diff_FI*GQG*diff_FI'*Ts;
P_l=diff_FI*P*diff_FI' + GQG;%P[k/k-1] est prête

% calcul de h, diff_h'
h=[x(1);x(2)];
diff_h=[1 0 0 0 0
         0 1 0 0 0];

%=====khaila=====
% system du filtre de kalman
K = P_l*diff_h'*inv(diff_h*P_l*diff_h'+R); % K[k]
out = x_l+K*(Y-h); % x[k] est prête
sys = out;
P = P_l-K*diff_h*P_l; % P[k] est prête

elseif flag == 3, % calcul des sorties
    sys = out;

elseif flag==4, % Calculer le temps de l'étape suivante
    sys=(round(t/Ts)+1)*Ts;
else
    sys=[];
end

```

- **Jeux des paramètres de la machine asynchrone utilisée**

Dans le cadre de la problématique liée à l'observation du flux, d'un machine asynchrone, tous les résultats présentés dans ce mémoire ont été obtenus sur deux machines différentes

La première est une machine à rotor à cage: M1

- Caractéristiques :

Puissance=0.75 kW Facteur de puissance=0,75 Courant=3,5 A Tension=380 V

Couple=5 Nm Fréquence=50 Hz Vitesse=1430 tr/mn

- Paramètres électriques :

$R_s=4.35 \Omega$ $R_r=2.48 \Omega$ $L_r=M_{sr}=176 \text{ mH}$ $L_s=200 \text{ mH}$

- Paramètres mécaniques :

$J=5.4 \text{ mN.m.rad-2.s}^2$ $p=2$ $F_v=1.6 \text{ mN.m.rad-1.s}$

La deuxième est une machine à rotor de Trois-Rivières : M2

- Caractéristiques :

Puissance=175 W Courant=1.2 Arms Tension=380 V

Couple=5 Nm Fréquence=50 Hz Vitesse=1670RPM

- Paramètres électriques :

$R_s=12.5 \Omega$ $R_r=7.2 \Omega$ $M_{sr}=477.5 \text{ mH}$ $L_r=L_s=499.25 \text{ mH}$

- Paramètres mécaniques :

$J=0.0022 \text{ Kg.m-1}$ $p=2$ $F_v=0.001224 \text{ Kg.s-1}$